STARWARPÒ

Reference Guide

Version 6.1.0

Order Number: STWRG610

serena

STARWARP$^{\grave{0}}$ 6.1.0 Reference Guide

First Edition – January, 1, 2000

STARWARP$^{\grave{0}}$ Publications

Additional copies of the following STARWARP publications may be purchased from SERENA:

STARWARP 6.1.0 User Guide is a guide to STARWARP concepts and facilities.

- The User Guide section explains how StarWarp can help with Year 2000 conversions and is a guide to StarWarp functions and features.
- The Messages section explains both online and batch messages for StarWarp. This section also describes how to deal with error situations.
- The StarBat section contains extensive information on batch facilities. These facilities are designed to execute as background programs and to process control statements. Programmers debugging or designing batch facilities background jobs use the StarBat section.

StarWarp 6.1.0 Reference is designed for professional applications and systems programmers who have experience with programming, file structures, utilities, and testing practices within the mainframe environment.

- The Reference section is an alphabetic guide to the functions and commands of StarWarp.
- The Installation section explains procedures for setting up StarWarp, including installation tailoring, execution of the program, and installation verification.
- The StarWarp Appendixes section provides a history of the various upgrades of StarWarp.

Special Notices

About SERENA Software

SERENA Software specializes in providing integrated products and services to support enterprise software change management. SERENA focuses on providing a total solution to the problems of locating, fixing, and managing changes to applications and systems software. SERENA Software's products have become industry-recognized, and are currently used in more than 2,000 data centers worldwide.

SERENA is headquartered in Burlingame, California, and has offices in the United Kingdom, Germany, and Canada.

FULL.CYCLE™ Mainframe

Change Man® *For Software Life Cycle Management*

A comprehensive software change management system that automates the entire software change life cycle.

Change Transfer™ *For Efficient VSAM Backups*

Through its innovative fingerprinting process, identifies and backs up only those VSAM records that have been changed, reducing the need for daily backups.

Comparex® *For Intelligent Comparison*

The industry-leading tool for efficient application testing and software quality assurance. It is used for regression testing, auditing, reconciliation of entire production libraries or of single members, and for data extraction for testing.

Detect+Resolve™ Mainframe (SyncTrac®) *For Intelligent Change Detection and Synchronization*

A synchronization and data set change tracking product for the MVS software environment. Primary applications include centralized migration of systems maintenance, systems audit and change control, disaster recovery, and data center consolidation or migration.

Merge+Reconcile™ Mainframe (CDF™) *For Intelligent Version Merging and Reconciliation*

An MVS product that consolidates multiple versions of application programs, primarily for collaborative development. It quickly compares and intelligently merges ongoing internal development and third party software code into one consolidated file.

SERNET™ Connect (X:Change™) *For Enterprise Connectivity*

A state-of-the-art tool that provides cross-platform file management support for programmers involved in desktop-based software development for mainframe and client/server applications. SERNET Connect comprises powerful features that enable users to synchronise, transfer, and distribute data and software quickly and easily.

SERNET™ Mainframe *The Core Enabling Infrastructure for FULL.CYCLE Mainframe*

StarTool® ***For File and Data Management***

An integrated programmer's environment comprising a workbench of productivity and diagnostic tools for managing sequential, partitioned, extended partitioned data sets, VSAM data sets, load modules, and other files.

StarWarp™ ***For Data Conversion***

A programmer's workbench for managing software conversions, manipulating dates, and managing data in most file types. StarWarp is a critical tool for use in Year 2000 and European Monetary Union (EMU) projects.

FULL.CYCLE™ Desktop

Detect+Resolve™ ***For Intelligent Software Configuration Problem Detection and Resolution***

A product that provides sophisticated and automated problem detection and resolution of software configuration problems for the enterprise desktop.

Getting Support

Support for the United States and Canada

Support Hours	Monday through Friday 5:00 A.M. to 5:00 P.M. Pacific time
Email	support@serena.com
URL	http://www.serena.com/html/support.htm
Phone	650 696-1850
Facsimile	650 696-1817
Mailing Address	SERENA Software, Inc. 500 Airport Boulevard, Second Floor Burlingame, CA 94010-1904

Support for the United Kingdom, Germany, Switzerland, and Austria

Support Hours	Monday through Friday 8:30 A.M. to 7:00 P.M. Central European time
Email	eurotech@serena.com
URL	http://www.serena.com/html/support.htm
Phone	44 1923 222233
Facsimile	44 1923 224433
Mailing Address	SERENA Software International, (UK) Ltd. 33 Metro Centre, Dwight Road Watford, Hertfordshire WD1 8SB, England

Support for All Other Countries

Please contact your local distributor.

Getting Support on the Internet

SERENA Software provides extensive support on the Internet through out “Customers Only” section on the Web and through our FTP server. To obtain an ID and password for access to support on the Internet, send a request by email to support@serena.com.

Once you have an ID and password, you can

- Report new issues.
- Search our problem tracking system for information about existing problems.
- View a knowledge base of frequently asked questions and helpful product hints.
- Query the call tracking database to obtain the current status of an open issue.
- Access our FTP server to download product fixes and online documentation in PDF format.

Also, as a SERENA customer, you can get the latest product information via email by subscribing to one of our mailing list servers (LISTSERV). For information on how to subscribe to the StarTool mailing list server, send an email message to support@serena.com.

Contents

Preface

This manual contains extensive information on the functions and commands of STARWARP in alphabetic order.

If you need assistance with STARWARP, please contact your marketing representative as noted on the cover page of this document or contact SERENA as noted in the STARWARP initialization messages.

STARWARP runs under the two major IBM MVS operating systems:

- MVS/ESA *(any release)*
- OS/390 (*any release*)

In addition, the following software environment should be available:

- ISPF and ISPF/PDF *(Version 4.1 or above)*
- TSO/E *(any release or any version)*

Version, Release, Modification

All software created and maintained by **SERENA** will have a Version, Release, and Modification level associated with it. Only when the Version or Release number changes *(usually annually)* will the full customer base *(those who are up to date with maintenance)* be issued new tapes and documentation; other releases can be requested if desired.

This manual describes:

```
STARWARP -- PDSTOOLS -- PDS/E
Version .............. 6
Release .............. 1
Modification ......... 0
Julian release date ... 2000.001
```

Subcommands and Functions Table

Subcommand	Short	Function	Reference	Source	Load
ATTRIB	A	List and modify member attributes	page 7	yes	yes
BLK3380	BLK338	Optimize 3380 disk utilization	page 14	yes	yes
BLK3390	BLK339	Optimize 3390 disk utilization	page 16	yes	yes
BLK9345	BLK9	Optimize 9345 disk utilization	page 18	yes	yes
BROWSE	B	Browse data	page 20	yes	yes
CALC	CAL	Floating point calculator (ISPMODE only)	page 22	yes	yes
CAX	CAX	List active system catalogs (ISPMODE only)	page 23	yes	yes
CHANGE	C	Switch to a different data set	page 26	yes	yes
CMDTBL	CMD	Manage ISPF command tables	page 28	yes	yes
COMPARE	CO	Display data differences	page 32	yes	yes
COMPDIR	COMPD	Compare member directory entries	page 33	yes	yes
CONDEND	COND	Conditionally terminate STARWARP	page 37	yes	yes
CONTROL	CON	Control STARWARP global options	page 38	yes	yes
CREATE	CR	Create a data set based on the current data set	page 44	yes	yes
CSECTS	CS	Map module CSECTs (ISPMODE only)	page 47	no	yes
DCF	DC	Script a data set	page 51	yes	no
DDNAME	DD	List TSO session allocations (ISPMODE only)	page 52	yes	yes
DECODE	DEC	Decrypt a member	page 57	yes	no
DELINK	DELI	Produce object code from a load module	page 58	no	yes
DIRENTRY	DIR	Interpret member directory entries	page 59	yes	yes
DISASM	DISA	Disassemble load modules	page 60	no	yes
DISPLAY	D	List member names from the directory	page 62	yes	yes
DSAT	DSA	Display data set attributes	page 64	yes	yes
DSNAME	DS	Display data set allocation information	page 68	yes	yes
DVOL	DV	Display volume attributes and statistics	page 69	yes	yes
EDIT	E	Edit data	page 71	yes	no
EDREC	EDR	Invoke ISPF edit recovery	page 74	yes	no
ENCODE	ENC	Encrypt a member	page 75	yes	no
END	EN	Terminate the current function or STARWARP	page 76	yes	yes
EXCLUDE	EXC	Drop members from MEMLIST	page 77	yes	yes
EXEC	EX	Execute a CLIST with subcommands	page 82	yes	yes
FILTER	FIL	Filter data sets for LISTC/LISTF and MASK	page 83	yes	yes
FIND	FI	Search for a string	page 85	yes	yes
FINDMOD	FINDM	Locate system modules	page 94	yes	yes

Subcommand	Short	Function	Reference	Source	Load
FSE	FS	Edit data	page 96	yes	no
GO	GO	Switch STARWARP sessions (ISPMODE only)	page 98	yes	yes
HELP	H	Provide information on using STARWARP	page 100	yes	yes
HEX	HEX	Hexadecimal calculator (ISPMODE only)	page 101	yes	yes
HISTORY	HI	Display CSECT IDR data from a module	page 102	no	yes
IDCAMS	IDC	Invoke an IDCAMS command	page 111	no	no
IF	IF	Search for members with desired attributes	page 113	yes	yes
ISPF	ISPF	Stack an ISPF session	page 120	yes	yes
ISPMODE	ISPM	Switch STARWARP to an ISPF display mode	page 121	yes	yes
ISPXEQ	ISPX	Invoke STARWARP with an ISPMODE option	page 122	yes	yes
LIST	LI	Display data	page 123	yes	yes
LISTA	LA	List TSO session allocations (ISPMODE only)	page 130	yes	yes
LISTC	LC	List data sets from a catalog (ISPMODE only)	page 135	yes	yes
LISTF	LF	List data sets from a volume (ISPMODE only)	page 144	yes	yes
LISTGRP	LISTG	Display the member group names	page 151	yes	yes
LISTV	LV	List disk volumes (ISPMODE only)	page 152	yes	yes
LOG	LO	Select one of 9 session logs (ISPMODE only)	page 158	yes	yes
MAP	MA	Display the CSECT structure of a module	page 163	no	yes
MASK	MAS	Front-end LISTC and LISTF (ISPMODE only)	page 166	yes	yes
MEMBERS	ME	List member names in a member group	page 168	yes	yes
MEMLIST	ML	Build a member list table (ISPMODE only)	page 169	yes	yes
NUCMAP	NUC	Map nucleus (ISPMODE only)	page 182	yes	yes
OPTIONS	O	Provide a directory of subcommands	page 185	yes	yes
PATTERN	P	Display member names from the directory	page 186	yes	yes
PBROWSE	PB	Browse data (ISPMODE only)	page 188	yes	yes
PGMDOC	PGM	List module descriptions	page 191	no	yes
PRINT	PR	Print a hardcopy listing	page 192	yes	yes
PROFMAN	PRO	Manage saved tables (ISPMODE only)	page 193	yes	yes
QUIT	Q	Terminate STARWARP	page 197	yes	yes
READOBJ	READ	Disassemble object code	page 198	yes	no
RECALL	R	Display the previous subcommand	page 200	yes	yes
REPLACE	REPL	Change member contents	page 201	yes	yes
REVIEW	REV	Browse a member without ISPF	page 210	yes	yes
SUBLIST	SUBL	Create a member subset	page 211	yes	yes
SUBMIT	SUB	Submit data for background processing	page 213	yes	no
STATUS	ST	Display the status of ISPMODE functions	page 214	yes	yes
SVCMAP	SVC	Investigate SVC routines	page 216	yes	yes

Subcommand	Short	Function	Reference	Source	Load
TSO	T	Invoke a TSO command processor or CLIST	page 219	yes	yes
TSOEDIT	TSOE	Edit data	page 220	yes	no
TSOLIST	TSOL	List data	page 222	yes	yes
USAGE	U	Display data set statistics	page 223	yes	yes
VERIFY	V	Validate members or a data set	page 227	yes	yes
VPRINT	VP	Print on a VTAM printer	page 233	yes	yes
VTOC	VT	Display data sets on disk volumes	page 235	yes	yes
VUSE	VU	Display disk usage and statistics	page 241	yes	yes
WHOHAS	WH	Check allocation status of a data set	page 244	yes	yes
WORKPAD	W	Store or execute commands (ISPMODE only)	page 245	yes	yes
XREF	XR	Display intra-module references	page 251	no	yes

General

Legend

This legend describes the symbols and abbreviations used in the following chapters.

[ ] Brackets enclose an optional entry.

( ) Parentheses must be coded as shown in the examples.

{ } Braces indicate a required entry when more than one selection is available.

/ Slash indicates an alternate entry; it is commonly used to indicate a choice of keywords or operands.

__ Underscores indicate a default value.

CAPS Uppercase letters indicate a keyword or name to be entered as shown.

lowercase Lowercase letters indicate that variable information is to be supplied.

memgroup memgroup is always the first operand in the syntax of the following subcommands where it appears. It must normally be entered (for example, you may use *****) or an entry assist panel will be provided to prompt for subcommand parameters. By convention, any subcommand whose first operand is defined as a **memgroup** can define and manipulate subgroups of members.

Note: the **memgroup** specification is always *omitted* for VSAM, sequential or direct data sets.

Organization

Most of the rest of this manual has an underlined topic title at the top of each page and each chapter is generally organized alphabetically so that you can find topics by searching the topic titles. Following are some guidelines on how to find information on various types of functions and commands.

Functions See **Subcommands and Functions** on page 6 for documentation on all subcommands and functions; see the **Subcommands and Functions Table** on page ii for a directory and short description of all subcommands and functions; see Table of Contents entry "Subcommands and Functions"; and see Index entry "Function" and Index entries for individual function names.

Subcommands Same as "Functions" above except that Index entry "Subcommand" provides a directory of all subcommands and functions.

Commands See **All Commands** on page 256 for documentation on all ISPMODE commands; see Index entry "Command" for a directory of all commands; for more specific information, see the documentation for the related function.

Line commands See Index entry "Line command" for a directory of all line commands; for more specific information, see the documentation for the related function.

Block line commands See Index entry "Block line command" for a directory of all block line commands; for more specific information, see the documentation for the related function.

Global commands See **All Commands** on page 256 for documentation on all ISPMODE commands (including global commands); see Index entry "Global command" for a directory of global commands; for more specific information, see the documentation for the related function.

STARWARP Command

Purpose The STARWARP command accesses and manipulates data objects, including:

- partitioned data sets (PDS or PDSE data sets)
  - source members -- RECFM=F or RECFM=V
  - load members -- RECFM=U
  - member lists
  - subsets or groups of members
- sequential, direct and VSAM data sets

Example STARWARP lib.cntl

Syntax

```
STARWARP {dsn [VOLUME(vser)/VOLSET(vdef)] [ SHR/OLD]
           / * / FILE(ddname) [NUMBER(num)]}
         [ISPMODE / XISPMODE / MEMLIST memg /
           / ISPXEQ cmnd / subcommand ]
```

Aliases PDS, PDSE, PDSTOOLS, STARTOOL

Defaults ISPMODE if in an ISPF session, XISPMODE otherwise; SHR

Required **dsn** or FILE(ddname)

Operands

dsn identifies the data set name. If the data set name is not entered in quotes ('), your TSO PREFIX will be appended to the start of the entered data set name. If ***** is entered in this position, **FILE(ISPPROF)** will be assumed.

Note that if your system has password security, you may enter your data set password after the data set name and a slash (/). The syntax is: **dsn/password**

VOLUME(vser) specifies the volume name on which the data set resides. If a volume name is entered, SYSALLDA is assumed as the unit name; otherwise, the unit name from the catalog is used.

Note: this parameter should be used if the data set is not cataloged or if the catalog entry is not to be used. To use the catalog for the STARWARP command even if VOLSET is in effect, you may enter **VOLUME(*)**.

VOLSET(vdef) specifies a default volume name for data set references. For example, if **VOLSET(NEWRES)** were entered, subsequent CHANGE subcommands with a data set name and no VOLUME parameter will assume **VOLUME(NEWRES)**.

Note: this parameter should be used if a default volume is desired. To nullify the effect of VOLSET, enter **VOLSET(*)** on a subsequent CHANGE subcommand.

SHR allocate the data set with a disposition of SHR; allow simultaneous use of this data set by others. The use of **SHR** is recommended.

OLD allocate the data set with a disposition of OLD; do not allow simultaneous use of this data set by others. The use of **SHR** is recommended.

FILE(ddname) identifies the DDNAME of a preallocated data set. Note that only disk data sets (including VIO) are supported. If the FILE keyword is used, **dsn**, SHR/OLD, VOLSET and VOLUME should not also be used. However, if the data set is concatenated, the data set is reallocated so SHR or OLD may be used.

NUMBER(num) specifies the concatenation number desired for the DDNAME allocation for the FILE keyword. Note that num defaults to 1 but if num is larger than the number of concatenated data sets, the last data set in the sequence will be used.

ISPMODE requests that STARWARP initialize in ISPMODE. Note that this is the initial mode only; ISPMODE and MEMLIST may be suspended with a later subcommand.

XISPMODE requests that STARWARP initialize in line mode (independent of ISPF). Note that this is the initial mode only; if desired at a later time, an ISPMODE or MEMLIST subcommand may be entered.

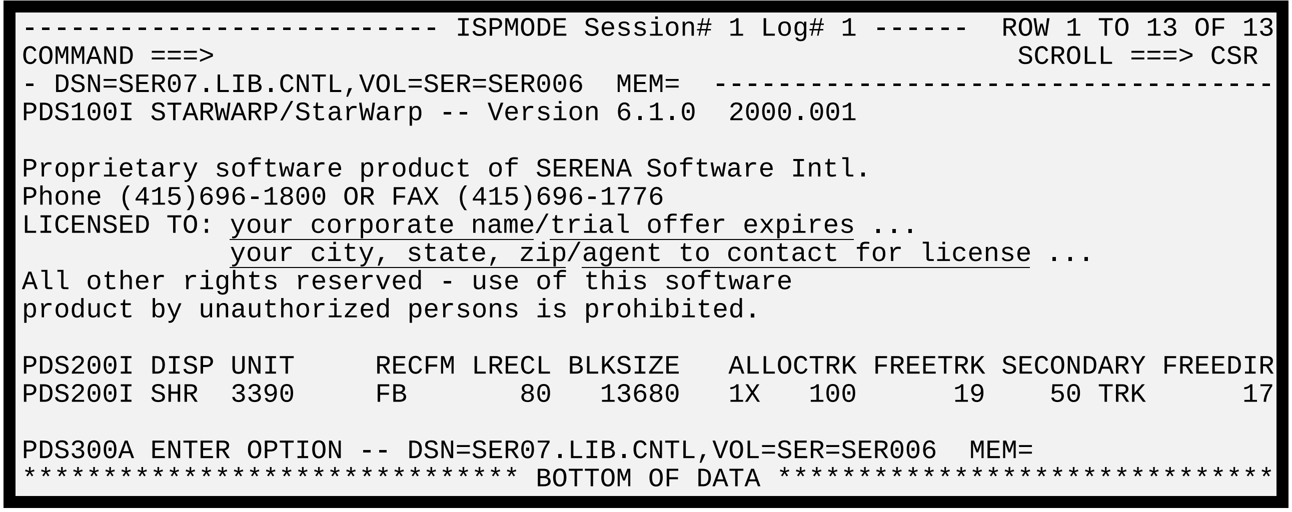
MEMLIST memg requests that STARWARP initialize with a member list set to the **memg** member group. Note that this is the initial mode only; MEMLIST and ISPMODE may be suspended with a later subcommand.

ISPXEQ cmnd requests that STARWARP initialize with the ISPMODE service (usually DDNAME, LISTA, LISTC, LISTF or LISTV) named. Note that this is the initial mode only; other STARWARP subcommands may be entered after the initial display.

subcommand requests that STARWARP perform the single subcommand and terminate. This is a special subcommand mode for performing only a single subcommand. In this mode, ISPF services are not available and STARWARP operates in line mode only. Also, no YES/NO prompting is provided; instead, YES responses are assumed.

When STARWARP is invoked in single subcommand mode, the return code is set to the numeric value of the first warning or error message encountered.

Remarks The following sample screen is the result of entering **TSO STARWARP lib.cntl** from ISPF.



```
------------------------- ISPMODE Session# 1 Log# 1 ------  ROW 1 TO 13 OF 13
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=  ---------------------------------
PDS100I STARWARP/StarWarp -- Version 6.1.0  2000.001

Proprietary software product of SERENA Software Intl.
Phone (415)696-1800 OR FAX (415)696-1776
LICENSED TO: your corporate name/trial offer expires ...
             your city, state, zip/agent to contact for license ...
All other rights reserved - use of this software
product by unauthorized persons is prohibited.

PDS200I DISP UNIT     RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390     FB       80   13680   1X   100      19    50 TRK      17

PDS300A ENTER OPTION -- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=
****************************** BOTTOM OF DATA ******************************
```

Figure 1. Sample STARWARP Command

STARWARP subcommands are available to perform many different functions. There are currently over 90 different subcommands, several of which have multiple operands. For additional information, see the next chapter, **Subcommands and Functions** on page 6.

Hexadecimal Member Names

Member names can be entered in hexadecimal for any of the subcommands which accept member names (or partial member names) as input. Thus, for these subcommands, **x'd7c4e2c5'** and **PDSE** are equivalent. Also, note that **x'333'** and **x'0333'** are equivalent.

Default Member Names

If ***** is entered in the member group position for an ATTRIB, BROWSE, COMPDIR, DCF, DELINK, DIRENTRY, DISASM, EDIT, EXCLUDE, FIND, FSE, HISTORY, IF, LIST, MAP, MEMBERS, MEMLIST, PGMDOC, PRINT, READOBJ, REPLACE, REVIEW, SPFEDIT, SUBLIST, SUBMIT, TSOEDIT, TSOLIST, VERIFY, VPRINT or XREF subcommand, the STARWARP command will use the last member name or member group entered for any of these subcommands.

When a default member name or group is specified by one of the above subcommands, the member name or group becomes the "current member group" or the "default member name or group" (the ALIAS subcommand also modifies the default member group but it does not process member groups). The current member group may be referred to by this set of subcommands until it is again redefined.

Member Groups

The above subcommands which allow default member names also accept ranges, patterns, combinations and lists of member names as shown below; for more information, see **Appendix A. Member Name Forms** on page 261.

1. A single asterisk (*****) may be entered to refer to the current member group for any of these subcommands.
2. A member name range is of the form **name1:name2**, which selects members based on character string ranges in their names.
3. A pattern member name is of the form **name1/name2**, which selects members based on character string matches in their names.
4. A combination member name is of the form **name1*name2**, which selects members based on the range name1:name1 and on the pattern name2 in the remainder of the member name.
5. A placeholder is allowed in a simple member name, a pattern member name or in a combination member name. This is specified by using a ? or % anywhere in the member name and it acts as a single character "wild card".
6. A member name list is a parenthesized list of member name specifications. It may contain normal member names as well as the member name forms described just above. When a member list is used, the subcommand usually receives control separately for each member specification in the list.
7. A single equal symbol(=) may be entered to refer to the list of members in the current MEMLIST; if a MEMLIST is not active, the equal symbol is equivalent to a asterisk (the current member group).

Abbreviations

Subcommand names can be abbreviated to the first few characters of the subcommand name. For additional information on subcommand name abbreviations, see the **Subcommands and Functions Table** on page ii. Keyword operands may also be abbreviated. The general rule is that you may abbreviate by dropping ending characters of a keyword name as long as the keyword is unique among other keywords for the same subcommand.

Subcommands and Functions

This section documents all subcommands and functions in alphabetical order. For an alphabetical list of all subcommands, see the **Subcommands and Functions Table** on page ii; you can also locate subcommands by referring to the underlined subcommand or function name on the top of each page in this chapter.

Subcommands can be thought of as a request for a operation that would be supported in all StarWarp environments.

A function command is a request for an ISPMODE operation. Most functions support their own set of commands; functions commands control ISPF tables which are available throughout a StarWarp session in parallel mode.

While operating in a StarWarp session, commands are ISPMODE-only requests for an operation.

Subcommands, function commands and commands, are entered as primary commands; that is, from the top command entry line of a panel. In most function tables, line commands may be entered in the "CMD" column; a line command is a request for an ISPMODE-only operation that affects only the table entry on which it is entered.

In each function which supports line commands, **=** and **X** are generally available as line commands. A **=** line command may be used to repeat the previously entered line command and **X** line commands may be used to drop table entries selectively. Either of these line commands may be followed by a number (for example, **=3** or **X3**) to operate on multiple table lines.

Another type of line command, block line commands, may be used to operate on multiple table entries. For example, the **XX** block line command can be used to mark table elements to drop from a table. **XX** operates on a range of entries; the first table element with an **XX** command is considered the first entry in the range; all following elements are dropped up to a paired entry with another **XX** block line command. **XX** and **==** block line commands are available in each function that supports line commands.

Commands, line commands and block line commands are documented in this section but only under the function in which they are supported. For a discussion of all ISPMODE commands, see **All Commands** on page 256.

ATTRIB Subcommand (for source members)

Purpose The ATTRIB subcommand lists and modifies the attributes of a member. If the attributes of a group of members is being updated, STARWARP will identify the members to be updated and ask whether or not to continue with the update. ISPF statistics or SSI information can be added or modified. Since many of the operands apply either to load members or source members only, ATTRIB is described separately with each set of operands.

Example ATTRIB mema:memb id(duserid)

Syntax

```
ATTRIB   memgroup [ALIASINFO/ NOALIASINFO ]
                  [ALIAS/ NOALIAS          ]
                  [LASTREAD               ]
                  [ADDSTATS               ]
                  [ID(who)/USERID(who)    ]
                  [MOD(mm)                ]
                  [NONE                   ]
                  [SSI(hxdata)/NOSSI      ]
                  [UNALIAS                ]
                  [VER(nn)                ]
                  [CREATED(yyyy/mm/dd)    ]
                  [MODIFIED(yyyy/mm/dd)   ]
                  [TIME(hhmm)             ]
                  [SIZE(nnnnn)/RESIZE     ]
                  [INIT(nnnnn)            ]
                  [MODI(nnnnn)            ]
```

Aliases A, AT, ATT, ATTR, ATTRI, ATTRIB

Defaults for alias members: memgroup, ALIASINFO depending on the CONTROL setting, NOALIAS
for other members: memgroup, NOALIASINFO, NOALIAS

Required none

Operands -- source members

memgroup identifies the source member(s) whose attributes are to be displayed or updated.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

ALIASINFO provide the names of any alias or main members--this operand does not change member attributes.

NOALIASINFO do not provide any alias information--this operand does not change member attributes.

ALIAS also update any associated members (aliases, main members or apparent aliases).

NOALIAS do not also update any associated members.

LASTREAD for members with ISPF statistics, format PDSMAN/MVS last read date information--this operand does not change member attributes.

ADDSTATS add ISPF statistics for members with no ISPF or SSI information.

ID(who) change the last modifier's userid (for members with ISPF statistics). Masking to retain current userid characters may be specified with the % character.

USERID(who) change the last modifier's userid (for members with ISPF statistics). Masking to retain current userid characters may be specified with the % character.

MOD(mm) change the ISPF modification level (for members with ISPF statistics).

NONE remove ISPF statistics or any SSI information.

SSI(hxdata) add or change the SSI information for the member. If less than 8 hexadecimal digits are entered, leading digits on the left will be assumed. Note that members with ISPF statistics may not also have SSI information.

NOSSI remove SSI information.

UNALIAS convert entry to a main member. Warning: if a main entry exists before this conversion, you will create an "apparent alias"; this situation can be detected by a VERIFY subcommand or ATTRIB with the ALIASINFO keyword.

VER(nn) change the ISPF version number (for members with ISPF statistics).

CREATED(yyyy/mm/dd) change the ISPF creation date (for members with ISPF statistics).

MODIFIED(yyyy/mm/dd) change the ISPF modification date (for members with ISPF statistics).

TIME(hhmm) change the ISPF modification time (for members with ISPF statistics).

SIZE(nnnnn) change the ISPF member size (for members with ISPF statistics).

RESIZE recalculate the member size (for members with ISPF statistics).

INIT(nnnnn) change the ISPF initial size (for members with ISPF statistics).

MODI(nnnnn) change the ISPF modified size (for members with ISPF statistics).

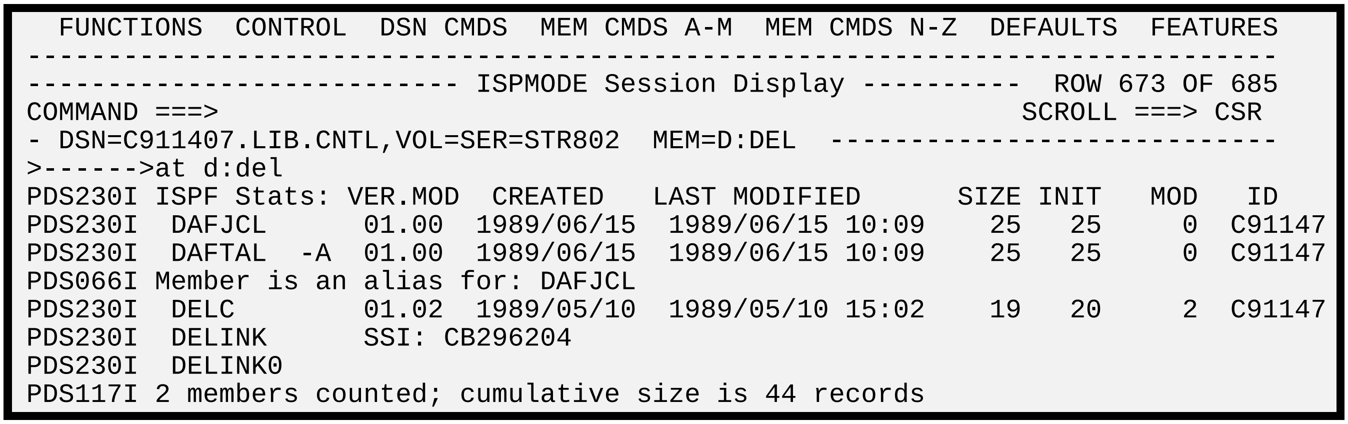
Remarks -- source members

If the member is an alias, **-A** is displayed after the member name. Note that an alias can be converted to a main member with the UNALIAS keyword.

If the keyword ALIASINFO is specified, the name of any associated members (aliases, the main member or any apparent aliases) will be displayed for each member. Otherwise, associated member information will be displayed for alias members if CONTROL ALIASINFO is in effect.

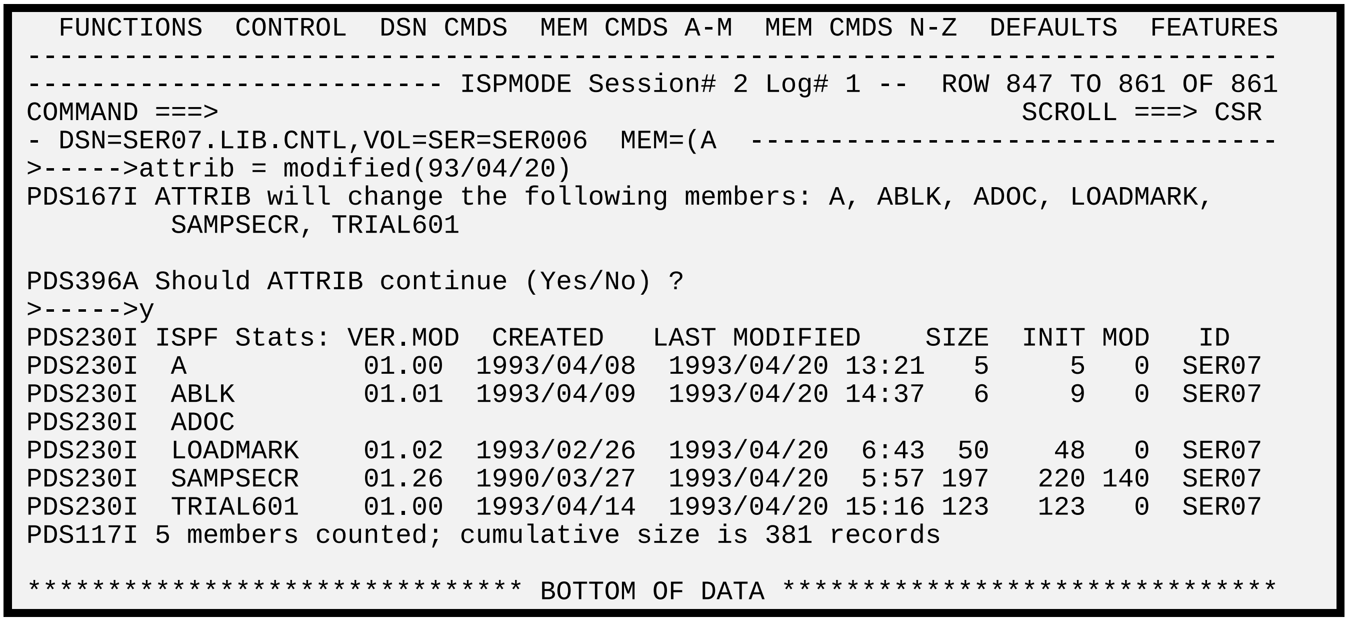
If the member has ISPF statistics or SSI information, this data may be deleted with the NONE keyword.

Caution: this subcommand modifies the data set if any attribute modifications are specified. To ensure data set integrity, you should allocate the data set as OLD or be aware of the data set update protection provided by the STARWARP command for SHR allocations (see **Appendix E. Update Protection** on page 271).



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 673 OF 685
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR802  MEM=D:DEL  ---------------------------
>------->at d:del
PDS230I ISPF Stats: VER.MOD  CREATED   LAST MODIFIED      SIZE INIT   MOD   ID
PDS230I  DAFJCL      01.00  1989/06/15  1989/06/15 10:09    25   25     0  C91147
PDS230I  DAFTAL  -A  01.00  1989/06/15  1989/06/15 10:09    25   25     0  C91147
PDS066I Member is an alias for: DAFJCL
PDS230I  DELC        01.02  1989/05/10  1989/05/10 15:02    19   20     2  C91147
PDS230I  DELINK      SSI: CB296204
PDS230I  DELINK0
PDS117I 2 members counted; cumulative size is 44 records
```

Figure 2. Sample ATTRIB Subcommand (source)



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 2 Log# 1 --  ROW 847 TO 861 OF 861
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(A  --------------------------------
>----->attrib = modified(93/04/20)
PDS167I ATTRIB will change the following members: A, ABLK, ADOC, LOADMARK,
         SAMPSECR, TRIAL601

PDS396A Should ATTRIB continue (Yes/No) ?
>----->y
PDS230I ISPF Stats: VER.MOD  CREATED   LAST MODIFIED    SIZE  INIT MOD   ID
PDS230I  A           01.00  1993/04/08  1993/04/20 13:21   5     5   0  SER07
PDS230I  ABLK        01.01  1993/04/09  1993/04/20 14:37   6     9   0  SER07
PDS230I  ADOC
PDS230I  LOADMARK    01.02  1993/02/26  1993/04/20  6:43  50    48   0  SER07
PDS230I  SAMPSECR    01.26  1990/03/27  1993/04/20  5:57 197   220 140  SER07
PDS230I  TRIAL601    01.00  1993/04/14  1993/04/20 15:16 123   123   0  SER07
PDS117I 5 members counted; cumulative size is 381 records

******************************* BOTTOM OF DATA ******************************
```

Figure 3. Sample ATTRIB Subcommand with update

ATTRIB Subcommand (for load members)

Purpose The ATTRIB subcommand lists and modifies the attributes of a member. If the attributes of a group of members is being updated, STARWARP will identify the members to be updated and ask whether or not to continue with the update.
Since many of the operands apply either to load members or source members only, ATTRIB is described separately with each set of operands.

Example ATTRIB mema:memb rent reus refr

Syntax

```
ATTRIB   memgroup [ALIASINFO/NOALIASINFO   ]
                  [ALIAS/NOALIAS           ]
                  [LKEDDATE/NOLKEDDATE     ]
                  [SHORT                   ]
                  [AMODE24/AMODE31/AMODEANY]
                  [AUTH/NOAUTH             ]
                  [DC/NODC                 ]
                  [EDIT/NOEDIT             ]
                  [ENTRY(name)             ]
                  [EXEC/NOEXEC             ]
                  [LOADONLY/NOLOADONLY     ]
                  [NONE                    ]
                  [PAGE/NOPAGE             ]
                  [REFR/NOREFR             ]
                  [RENT/NORENT             ]
                  [REUS/NOREUS             ]
                  [RLDFIX/NORLDFIX         ]
                  [RMODE24/RMODEANY        ]
                  [SSI(hxdata)/NOSSI       ]
                  [UNALIAS                 ]
```

Aliases A, AT, ATT, ATTR, ATTRI, ATTRIB

Defaults memgroup, ALIAS, LKEDDATE depending on the CONTROL setting,
ALIASINFO depending on the CONTROL setting

Required none

Operands -- load members

memgroup identifies the load member(s) whose attributes are to be displayed or updated.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

ALIASINFO provide the names of any alias or main members--this operand does not change member attributes.

NOALIASINFO do not provide any alias information--this operand does not change member attributes.

ALIAS also update any associated members (aliases, main members or apparent aliases).

NOALIAS do not also update any associated members.

LKEDDATE provide the date of the last linkage edit--this operand does not change member attributes.

NOLKEDDATE do not provide the date of the last linkage edit--this operand does not change member attributes.

SHORT provide output in an alternate short format--this operand does not change member attributes. This output format is used by default for THEN(attrib) or ELSE(attrib); for an example, see the documentation for message PDS232I. **Warning**: when the SHORT output format is in use, many standard ATTRIB module checks are not performed and an incorrectly created or modified load module may not be detected.

AMODE24 change addressing mode to 24.

AMODE31 change addressing mode to 31.

AMODEANY change addressing mode to ANY.

AUTH authorize the module with APF (equivalent to AC=1 in a linkage edit).

NOAUTH remove APF authorization (equivalent to AC=0 in a linkage edit).

DC indicate that the module is downward compatible with the level-E linkage editor.

NODC indicate that the module is not downward compatible with the level-E linkage editor.

EDIT allow linkage editing of the module.

NOEDIT disallow linkage editing of the module.

ENTRY(name) change the entry point address to the external symbol specified. Note: the symbol entered must be present in the load module.

EXEC allow execution of the module.

NOEXEC disallow execution of the module.

LOADONLY disallow use of the module except by a LOAD MACRO.

NOLOADONLY allow module use by any means.

NONE change attributes to the following: AMODE24, NOAUTH, NODC, EDIT, EXEC, NOLOADONLY, NOPAGE, NOREFR, NORENT, NOREUS, NOSSI and RMODE24.

Note: if any other parameters are coded with NONE, they will override any corresponding parameters.

PAGE require page alignment for the module.

NOPAGE allow any alignment for the module.

REFR add the refreshable attribute.

NOREFR remove the refreshable attribute.

RENT add the reentrant (and reusable) attribute.

NORENT remove the reentrant attribute.

REUS add the reusable attribute.

NOREUS remove the reusable attribute.

RLDFIX change the directory's RLD/CONTROL count to match the value in the first RLD entry (for modules linked since OS/VS).

NORLDFIX do not modify the directory's RLD/CONTROL count.

RMODE24 change residence mode to 24 (below the 16-Megabyte line).

RMODEANY change residence mode to ANY (above the 16-Megabyte line).

SSI(hxdata) add or change the SSI information for the member. If less than 8 hexadecimal digits are entered, leading digits on the left will be assumed.

NOSSI remove any SSI information for the member.

UNALIAS convert entry to a main member. Warning: if a main entry exists before this conversion, you will create an "apparent alias"; this situation can be detected by a VERIFY subcommand or ATTRIB with the ALIASINFO keyword.

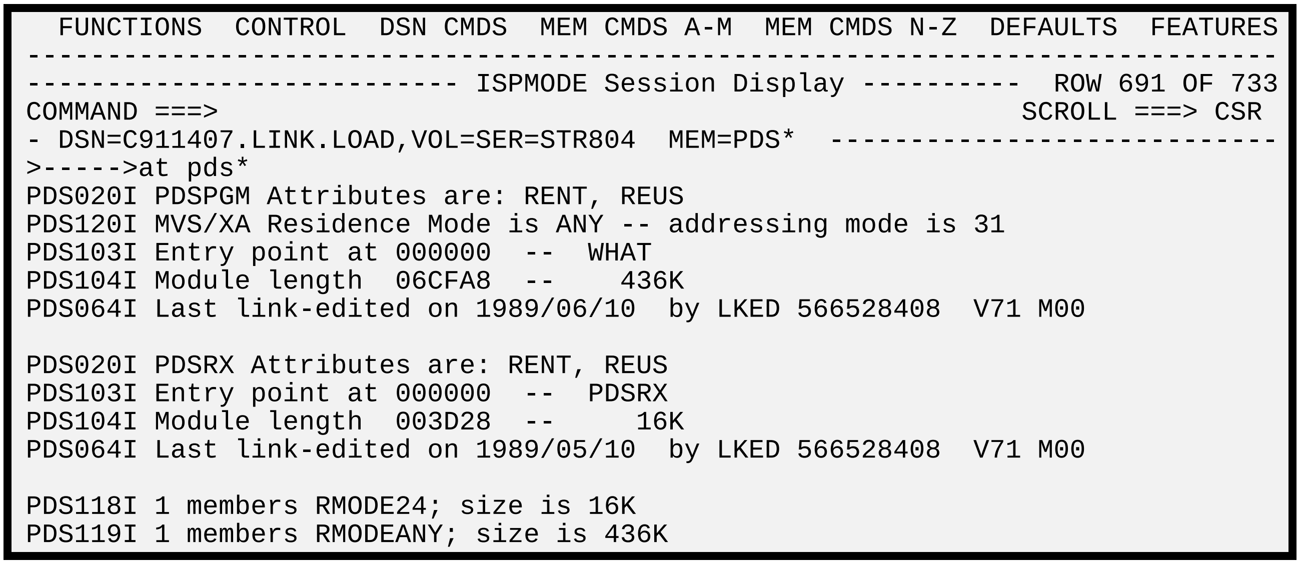
Remarks -- load members The following linkage editor attributes are identified if they are present for a member:

AMODE addressing mode (if AMODE is not 24 or RMODE is not 24).
AUTH APF authorized (or AC=1).
DC downward compatible (the module can be reprocessed by a Level-E linkage editor).
LOAD ONLY Only Loadable (OL -- the module can be brought into main storage only with a LOAD MACRO).
NOT EDIT Not Editable (the module cannot be linkage edited).
NOT EXEC Not Executable (the linkage editor noted an error and LET processing was not requested).
OVERLAY overlay load module structure.
PAGE page alignment required.
REFR refreshable (replaceable by a new copy during execution).
RENT reentrant (executable by several tasks simultaneously).
REUS reusable (executable by several tasks in serial order).
RMODE residence mode (if RMODE is not 24 or AMODE is not 24).
SCTR scatter load module structure.
SSI SSI information in hexadecimal.
TEST linked with the TEST option.

When the ATTRIB subcommand adds or deletes linkage attributes for a member it performs some additional processing based on the member's attributes and the keywords specified:

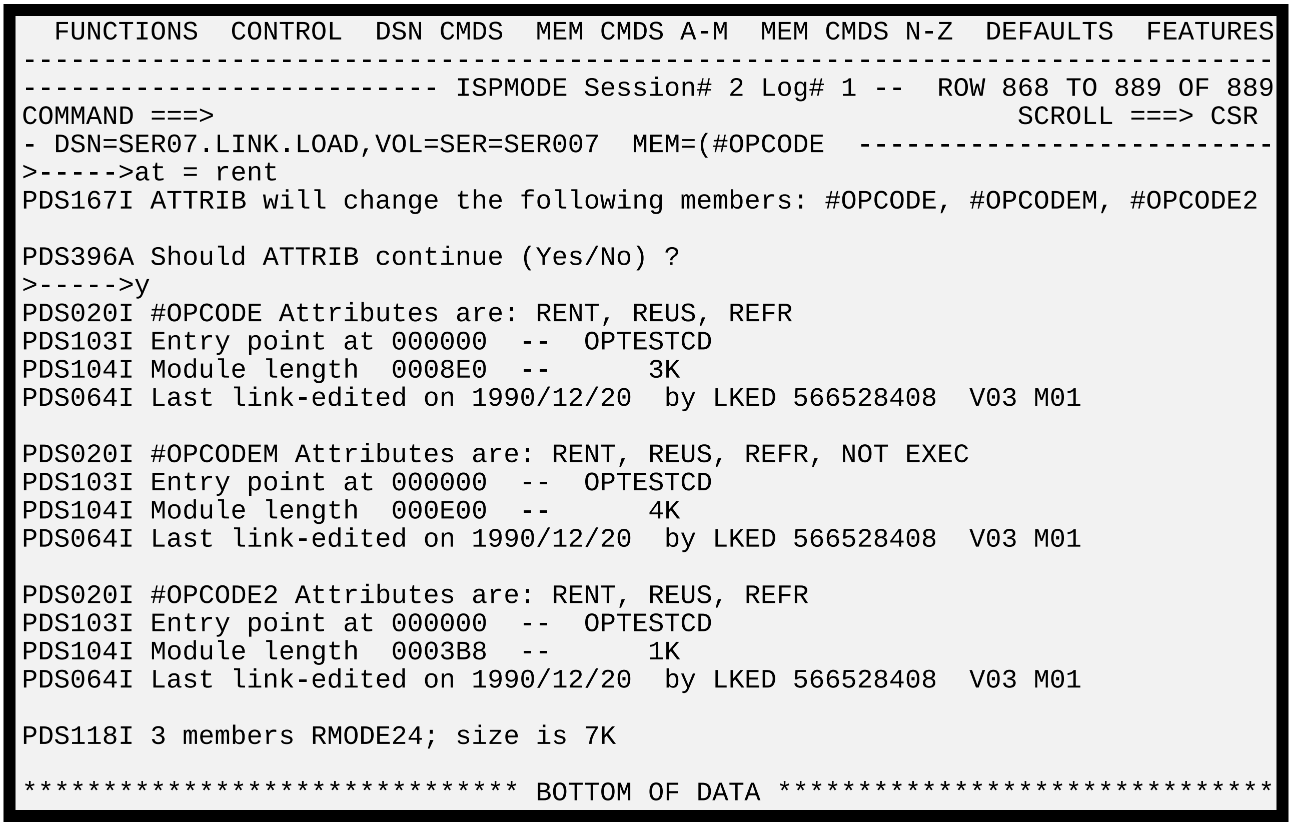
1. If a changed member is reentrant, the reusable attribute is also added (as does the linkage editor).
2. If a main member is assigned a RMODE or AMODE value, any associated alias directory entries are updated to reflect the same RMODE value and the main member's AMODE value.
3. Alias information is provided unless NOALIASINFO is specified or CONTROL NOALIASINFO is in effect.
   a) If the member is an alias, the name of any corresponding main member will be displayed if it exists; otherwise, the member name from the alias directory entry (the main member name at the time the alias was created) will be displayed.
   b) If the member is not an alias, the name of any corresponding alias members and any apparent aliases will be displayed.
4. If there is a mismatch between the directory RLD/CONTROL field and the first RLD entry of OS/VS linked modules, STARWARP displays an appropriate error message.

Caution: this subcommand modifies the data set if any attribute modifications are specified. To ensure data set integrity, you should allocate the data set as OLD or be aware of the data set update protection provided by STARWARP for SHR allocations. Ffor details, see **Appendix E. Update Protection** on page 271.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 691 OF 733
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=PDS*  ----------------------------
>----->at pds*
PDS020I PDSPGM Attributes are: RENT, REUS
PDS120I MVS/XA Residence Mode is ANY -- addressing mode is 31
PDS103I Entry point at 000000  --  WHAT
PDS104I Module length  06CFA8  --    436K
PDS064I Last link-edited on 1989/06/10  by LKED 566528408  V71 M00

PDS020I PDSRX Attributes are: RENT, REUS
PDS103I Entry point at 000000  --  PDSRX
PDS104I Module length  003D28  --     16K
PDS064I Last link-edited on 1989/05/10  by LKED 566528408  V71 M00

PDS118I 1 members RMODE24; size is 16K
PDS119I 1 members RMODEANY; size is 436K
```

Figure 4. Sample ATTRIB Subcommand (load)



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 2 Log# 1 --  ROW 868 TO 889 OF 889
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=(#OPCODE  --------------------------
>----->at = rent
PDS167I ATTRIB will change the following members: #OPCODE, #OPCODEM, #OPCODE2

PDS396A Should ATTRIB continue (Yes/No) ?
>----->y
PDS020I #OPCODE Attributes are: RENT, REUS, REFR
PDS103I Entry point at 000000  --  OPTESTCD
PDS104I Module length  0008E0  --      3K
PDS064I Last link-edited on 1990/12/20  by LKED 566528408  V03 M01

PDS020I #OPCODEM Attributes are: RENT, REUS, REFR, NOT EXEC
PDS103I Entry point at 000000  --  OPTESTCD
PDS104I Module length  000E00  --      4K
PDS064I Last link-edited on 1990/12/20  by LKED 566528408  V03 M01

PDS020I #OPCODE2 Attributes are: RENT, REUS, REFR
PDS103I Entry point at 000000  --  OPTESTCD
PDS104I Module length  0003B8  --      1K
PDS064I Last link-edited on 1990/12/20  by LKED 566528408  V03 M01

PDS118I 3 members RMODE24; size is 7K

******************************* BOTTOM OF DATA *******************************
```

Figure 5. Sample ATTRIB Subcommand with update

BLK3380 Subcommand

Purpose

The BLK3380 subcommand computes an optimal blocksize for a data set to be placed on a 3380 disk pack. The program output includes the following reports:

1. A summary blocksize report for the given LRECL and key length which includes the recommended blocksize to use.
2. A recommended data set space allocation.
3. A optional track capacity report for the provided key length.

Example

BLK3380 133

Syntax

```
BLK3380   lrecl   [KEYLENGTH(keylen)            ]
                  [TRACKCAP / NOTRACKCAP         ]
                  [BLKSIZE(blklen)              ]
                  [NUMBER(#rec) / RECORDS(#rec)  ]
                  [VERIFY                       ]
```

Aliases

BLK338, BLK3380

Defaults

KEYLENGTH(0), NOTRACKCAP, BLKSIZE(RECOMMENDED VALUE), NUMBER(100000)

Required

lrecl

Operands

lrecl the logical record length of the data which is to be placed in the data set.

KEYLENGTH(kl) the key length, in bytes, of the keys to be used in the data set. The maximum legal key length is 255.

TRACKCAP specifies a track capacity report is to be provided for the device using the specified (or default) key length.

Note that a track capacity report is also provided if NOTRACKCAP is not specified and lrecl exceeds the maximum blocksize for a track or BLKSIZE exceeds the maximum blocksize for a track.

NOTRACKCAP specifies that a track capacity report is not desired.

BLKSIZE(blklen) the blocksize to use for the allocation computation; if blklen is not entered (or zero is entered), the program's recommended blocksize will be used.

NUMBER(#rec) number of logical records that will be in the data set.

RECORDS(#rec) number of logical records that will be in the data set.

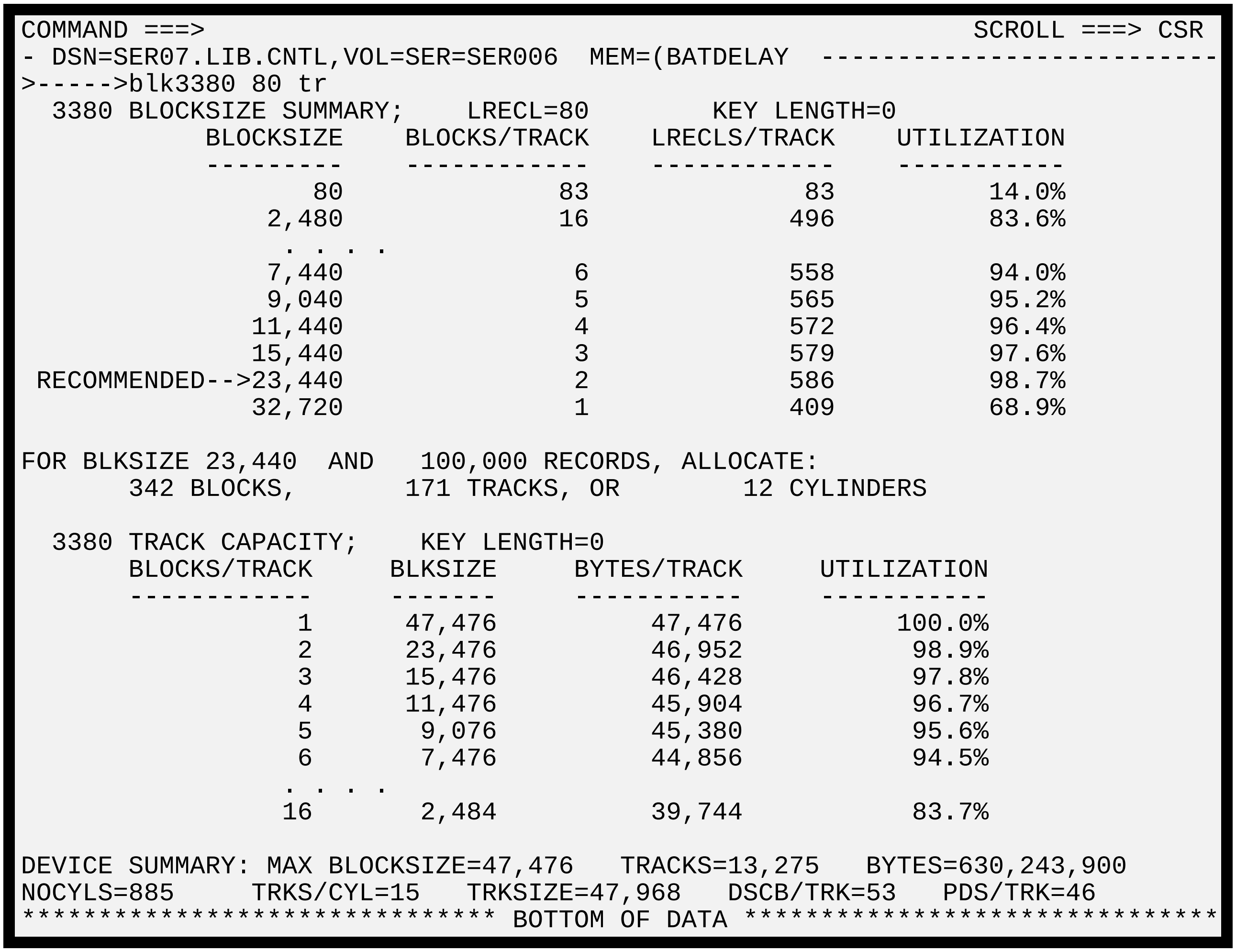
VERIFY specifies that the MVS "TRKCALC" routine is to be used to verify track capacity calculations. If VERIFY is used, the number of calls to "TRKCALC" to determine a track capacity table is output at the end of the output. With VERIFY on, a minimum of 34 calls is needed to determine a track capacity table; otherwise a minimum of 17 calls is needed to determine the track capacity table.

Remarks The recommended blocksize value is for data sets in which the predominant access is sequential; for data sets where random access time is critical or the usual access is random, a small blocksize (500-2000 bytes) should probably be used.

The recommended blocksize will usually tend to be near a half-track figure as this is considered to be the most efficient in terms of the trade-offs among buffer size, secondary storage requirements, channel use, number of input/outputs and overall processing time. This figure is only a general guide; for maximal efficiency considering other factors, study the generated blocksize summary report or a track capacity report.

The program's recommendations assume a fairly large amount of data is to be stored; data sets which occupy only a few tracks should probably be placed in partitioned data sets. In cases where this is not feasible, the use of a small blocksize (2400 - 4000 bytes) may be a good alternative practice.

This interface is optional, it should only be used if your installation has installed the public domain BLK3380 subcommand.



```
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  --------------------------
>----->blk3380 80 tr
  3380 BLOCKSIZE SUMMARY;    LRECL=80        KEY LENGTH=0
            BLOCKSIZE    BLOCKS/TRACK    LRECLS/TRACK    UTILIZATION
            ---------    ------------    ------------    -----------
                   80              83              83          14.0%
                2,480              16             496          83.6%
                 . . . .
                7,440               6             558          94.0%
                9,040               5             565          95.2%
               11,440               4             572          96.4%
               15,440               3             579          97.6%
 RECOMMENDED-->23,440               2             586          98.7%
               32,720               1             409          68.9%

FOR BLKSIZE 23,440  AND   100,000 RECORDS, ALLOCATE:
       342 BLOCKS,       171 TRACKS, OR        12 CYLINDERS

  3380 TRACK CAPACITY;    KEY LENGTH=0
       BLOCKS/TRACK     BLKSIZE     BYTES/TRACK     UTILIZATION
       ------------     -------     -----------     -----------
                  1      47,476          47,476          100.0%
                  2      23,476          46,952           98.9%
                  3      15,476          46,428           97.8%
                  4      11,476          45,904           96.7%
                  5       9,076          45,380           95.6%
                  6       7,476          44,856           94.5%
                 . . . .
                 16       2,484          39,744           83.7%

DEVICE SUMMARY: MAX BLOCKSIZE=47,476   TRACKS=13,275   BYTES=630,243,900
NOCYLS=885     TRKS/CYL=15   TRKSIZE=47,968   DSCB/TRK=53   PDS/TRK=46
******************************* BOTTOM OF DATA *******************************
```

Figure 6. Sample BLK3380 Subcommand

BLK3390 Subcommand

Purpose The BLK3390 subcommand computes an optimal blocksize for a data set to be placed on a 3390 disk pack. The program output includes the following reports:

1. A summary blocksize report for the given LRECL and key length which includes the recommended blocksize to use.
2. A recommended data set space allocation.
3. A optional track capacity report for the provided key length.

Example BLK3390 121

Syntax

```
BLK3390   lrecl   [KEYLENGTH(keylen)              ]
                  [TRACKCAP / NOTRACKCAP           ]
                  [BLKSIZE(blklen)                ]
                  [NUMBER(#rec) / RECORDS(#rec)  ]
                  [VERIFY                         ]
```

Aliases BLK339, BLK3390

Defaults KEYLENGTH(0), NOTRACKCAP, BLKSIZE(RECOMMENDED VALUE), NUMBER(100000)

Required lrecl

Operands

lrecl the logical record length of the data which is to be placed in the data set.

KEYLENGTH(kl) the key length, in bytes, of the keys to be used in the data set. The maximum legal key length is 255.

TRACKCAP specifies a track capacity report is to be provided for the device using the specified (or default) key length.

Note that a track capacity report is also provided if NOTRACKCAP is not specified and lrecl exceeds the maximum blocksize for a track or BLKSIZE exceeds the maximum blocksize for a track.

NOTRACKCAP specifies that a track capacity report is not desired.

BLKSIZE(blklen) the blocksize to use for the allocation computation; if blklen is not entered (or zero is entered), the program's recommended blocksize will be used.

NUMBER(#rec) number of logical records that will be in the data set.

RECORDS(#rec) number of logical records that will be in the data set.

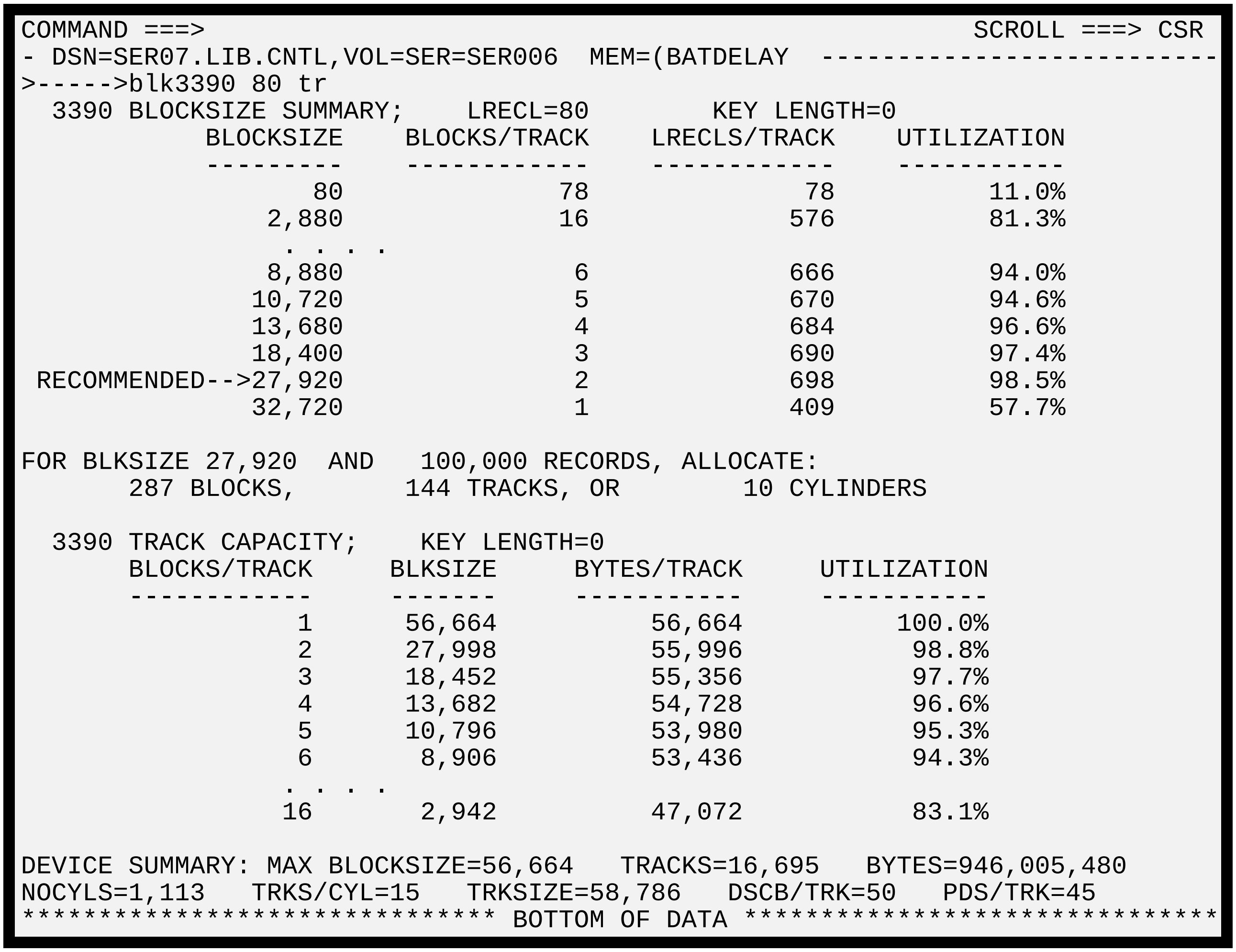
VERIFY specifies that the MVS "TRKCALC" routine is to be used to verify track capacity calculations. If VERIFY is used, the number of calls to "TRKCALC" to determine a track capacity table is output at the end of the output. With VERIFY on, a minimum of 34 calls is needed to determine a track capacity table; otherwise a minimum of 17 calls is needed to determine the track capacity table.

Remarks The recommended blocksize value is for data sets in which the predominant access is sequential; for data sets where random access time is critical or the usual access is random, a small blocksize (500-2000 bytes) should probably be used.

The recommended blocksize will usually tend to be near a half-track figure as this is considered to be the most efficient in terms of the trade-offs among buffer size, secondary storage requirements, channel use, number of input/outputs and overall processing time. This figure is only a general guide; for maximal efficiency considering other factors, study the generated blocksize summary report or a track capacity report.

The program's recommendations assume a fairly large amount of data is to be stored; data sets which occupy only a few tracks should probably be placed in partitioned data sets. In cases where this is not feasible, the use of a small blocksize (2400 - 4000 bytes) may be a good alternative practice.

This interface is optional, it should only be used if your installation has installed the public domain BLK3390 subcommand.



```
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  ---------------------------
>----->blk3390 80 tr
  3390 BLOCKSIZE SUMMARY;    LRECL=80        KEY LENGTH=0
            BLOCKSIZE    BLOCKS/TRACK    LRECLS/TRACK    UTILIZATION
            ---------    ------------    ------------    -----------
                   80              78              78          11.0%
                2,880              16             576          81.3%
                 . . . .
                8,880               6             666          94.0%
               10,720               5             670          94.6%
               13,680               4             684          96.6%
               18,400               3             690          97.4%
 RECOMMENDED-->27,920               2             698          98.5%
               32,720               1             409          57.7%

FOR BLKSIZE 27,920  AND   100,000 RECORDS, ALLOCATE:
       287 BLOCKS,       144 TRACKS, OR       10 CYLINDERS

  3390 TRACK CAPACITY;    KEY LENGTH=0
       BLOCKS/TRACK     BLKSIZE     BYTES/TRACK     UTILIZATION
       ------------     -------     -----------     -----------
                  1      56,664          56,664          100.0%
                  2      27,998          55,996           98.8%
                  3      18,452          55,356           97.7%
                  4      13,682          54,728           96.6%
                  5      10,796          53,980           95.3%
                  6       8,906          53,436           94.3%
                 . . . .
                 16       2,942          47,072           83.1%

DEVICE SUMMARY: MAX BLOCKSIZE=56,664   TRACKS=16,695   BYTES=946,005,480
NOCYLS=1,113   TRKS/CYL=15   TRKSIZE=58,786   DSCB/TRK=50   PDS/TRK=45
******************************* BOTTOM OF DATA ********************************
```

Figure 7. Sample BLK3390 Subcommand

BLK9345 Subcommand

Purpose

The BLK9345 subcommand computes an optimal blocksize for a data set to be placed on a 9345 disk pack. The program output includes the following reports:

1. A summary blocksize report for the given LRECL and key length which includes the recommended blocksize to use.
2. A recommended data set space allocation.
3. A optional track capacity report for the provided key length.

Example

BLK9345 80

Syntax

```
BLK9345   lrecl   [KEYLENGTH(keylen)             ]
                  [TRACKCAP / NOTRACKCAP          ]
                  [BLKSIZE(blklen)               ]
                  [NUMBER(#rec) / RECORDS(#rec)  ]
                  [VERIFY                        ]
```

Aliases

BLK9, BLK93, BLK934, BLK9345

Defaults

KEYLENGTH(0), NOTRACKCAP, BLKSIZE(RECOMMENDED VALUE), NUMBER(100000)

Required

lrecl

Operands

lrecl the logical record length of the data which is to be placed in the data set.

KEYLENGTH(kl) the key length, in bytes, of the keys to be used in the data set. The maximum legal key length is 255.

TRACKCAP specifies a track capacity report is to be provided for the device using the specified (or default) key length.

Note that a track capacity report is also provided if NOTRACKCAP is not specified and lrecl exceeds the maximum blocksize for a track or BLKSIZE exceeds the maximum blocksize for a track.

NOTRACKCAP specifies that a track capacity report is not desired.

BLKSIZE(blklen) the blocksize to use for the allocation computation; if blklen is not entered (or zero is entered), the program's recommended blocksize will be used.

NUMBER(#rec) number of logical records that will be in the data set.

RECORDS(#rec) number of logical records that will be in the data set.

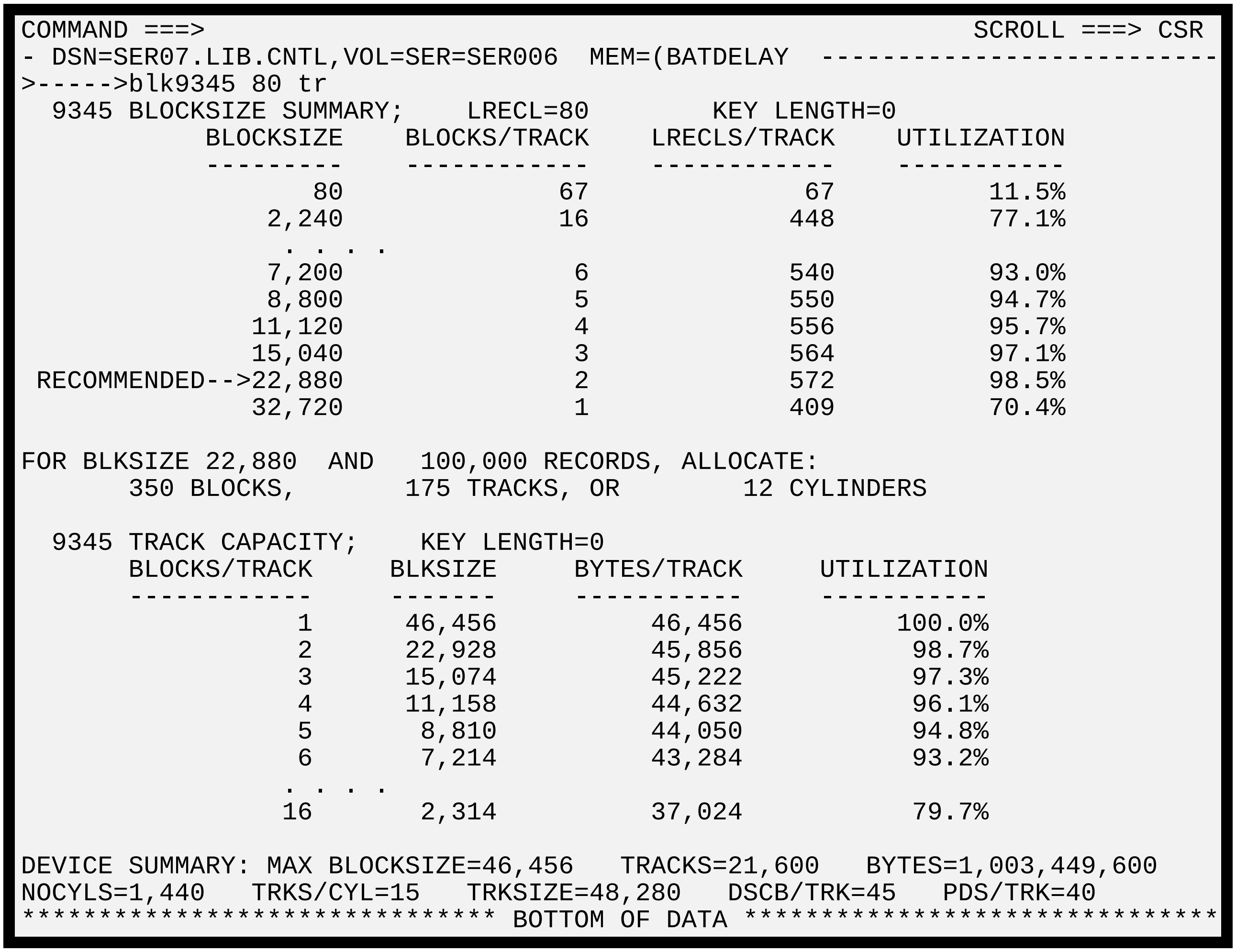
VERIFY specifies that the MVS "TRKCALC" routine is to be used to verify track capacity calculations. If VERIFY is used, the number of calls to "TRKCALC" to determine a track capacity table is output at the end of the output. With VERIFY on, a minimum of 34 calls is needed to determine a track capacity table; otherwise a minimum of 17 calls is needed to determine the track capacity table.

Remarks The recommended blocksize value is for data sets in which the predominant access is sequential; for data sets where random access time is critical or the usual access is random, a small blocksize (500-2000 bytes) should probably be used.

The recommended blocksize will usually tend to be near a half-track figure as this is considered to be the most efficient in terms of the trade-offs among buffer size, secondary storage requirements, channel use, number of input/outputs and overall processing time. This figure is only a general guide; for maximal efficiency considering other factors, study the generated blocksize summary report or a track capacity report.

The program's recommendations assume a fairly large amount of data is to be stored; data sets which occupy only a few tracks should probably be placed in partitioned data sets. In cases where this is not feasible, the use of a small blocksize (2400 - 4000 bytes) may be a good alternative practice.

This interface is optional, it should only be used if your installation has installed the public domain BLK9345 subcommand.



```
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  --------------------------
>----->blk9345 80 tr
  9345 BLOCKSIZE SUMMARY;    LRECL=80        KEY LENGTH=0
            BLOCKSIZE    BLOCKS/TRACK    LRECLS/TRACK    UTILIZATION
            ---------    ------------    ------------    -----------
                   80              67              67          11.5%
                2,240              16             448          77.1%
                 . . . .
                7,200               6             540          93.0%
                8,800               5             550          94.7%
               11,120               4             556          95.7%
               15,040               3             564          97.1%
 RECOMMENDED-->22,880               2             572          98.5%
               32,720               1             409          70.4%

FOR BLKSIZE 22,880  AND   100,000 RECORDS, ALLOCATE:
       350 BLOCKS,       175 TRACKS, OR        12 CYLINDERS

  9345 TRACK CAPACITY;    KEY LENGTH=0
       BLOCKS/TRACK     BLKSIZE     BYTES/TRACK     UTILIZATION
       ------------     -------     -----------     -----------
                  1      46,456          46,456          100.0%
                  2      22,928          45,856           98.7%
                  3      15,074          45,222           97.3%
                  4      11,158          44,632           96.1%
                  5       8,810          44,050           94.8%
                  6       7,214          43,284           93.2%
                 . . . .
                 16       2,314          37,024           79.7%

DEVICE SUMMARY: MAX BLOCKSIZE=46,456   TRACKS=21,600   BYTES=1,003,449,600
NOCYLS=1,440   TRKS/CYL=15   TRKSIZE=48,280   DSCB/TRK=45   PDS/TRK=40
******************************* BOTTOM OF DATA *******************************
```

Figure 8. Sample BLK9345 Subcommand

BROWSE Subcommand

Purpose The BROWSE subcommand enters ISPF browse for a member.

Example BROWSE   mema:memb

Syntax

```
BROWSE   memgroup
      * [FROMKEY(key)/FROMADDRESS(add)/FROMNUMBER(num)  ]
      * [TOKEY(key)/TOADDRESS(add)/TONUMBER(num)        ]

 *NOTE: Lines with an asterisk are supported for VSAM with PBROWSE or BRIF.
```

Aliases B, BR, BRO, BROW, BROWS, BROWSE

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults memgroup

Required none

Operands

memgroup identifies the member(s) to be browsed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access begins at the first record whose key matches (or is greater than) the portion of the key specified.

This parameter may be used with TOKEY and it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.

This parameter may be used with TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified.

This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.

This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

Remarks If **:** is entered for the member name position, an ISPF member selection list will be provided.

VSAM data sets are normally displayed using PBROWSE. A different interface may have been choosen during STARWARP installation. Of the other interfaces, only BRIF supports the positioning keywords such as FROMKEY. To determine which interface is used, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "BROWSE calls" with one of the following values:

BRIF this uses the ISPF BRIF interface which is up to 32760 characters wide
PBROWSE this uses PBROWSE services.
REVIEW this uses the REVIEW command (a non-ISPF fullscreen public domain program)
%VSAMMBR this uses MacKinney System's VSAM Utility Browse

If the ISPF BRIF interface is in use, STARWARP will keep the records of a data set in high region storage if the data set size does not exceed two megabytes or the data set is a path (since a path may not have unique keys). Records are provided to the BRIF service as requested; if you "max down" to the end of the data set, this could take a considerable amount of time. If the data set is in storage; however, subsequent positioning in the browse will be much faster.

CALC Function

Purpose The CALC command performs floating point calculations.

Example CALC

Syntax

```
CALC
```

Aliases CAL, CALC

Operands (no operands are supported for the CALC subcommand).

Remarks

The CALC ISPF table is displayed in response to a CALC command. When you are in a CALC display, you have many options: you may delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as a primary command.

The CALC table displays the calculator tape for your calculator. You can enter new calculations from the CALC display.

The following primary commands are supported directly for the CALC function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

EDITT[BL] (or **ET[BL]**) enters an edit session on CALC table data.

F finds a string and positions the display start location.
Syntax: `F anystring [ASIS]`
`[FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]`

OUT[PUT] outputs the CALC table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the CALC table based on a string match.
Syntax: `REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]`

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

X clears the CALC table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the CALC table; this is equivalent to **X ALL**

CAX Function

Purpose The CAX command displays the names and attributes of active system catalogs in an ISPF table.

Example CAX

Syntax

```
CAX
```

Aliases CAX, CAXW, CAXWA

Operands (no operands are supported for the CAX command).

Remarks

The CAX ISPF table is displayed in response to a CAX command. When you are in a CAX display, you have many options: you may delete a part of the table, sort the table in different directions, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

The CAX table displays the names of all active catalogs for your system, and allows you to enter LISTCAT-type line commands against them.

The catalogs are open at the time of the list, and it is possible to use the information displayed in fixing suspected broken catalogs. STARWARP does not fix catalogs, but the display of information on the open catalogs may aid in the diagnosis of the problems.

The results of line commands against the CAX list are usually captured in the STARWARP log. They can possibly be saved for later session use in one of the alternate logs.

The CAX table displays the active system catalogs and information about each catalog on a line level: Type, Status, CAXWA address, relative entry number, volume name and data set name.

Type The type of catalog

- **ICF** Integrated catalog facility type catalog.
- **VSAM** Old type VSAM catalog.
- **RECV** Old type VSAM recoverable catalog.
- **MSTR** Master catalog.

Status The status of the catalog.

- **ACTIVE** Usual catalog status.
- **DELETE** A catalog delete has been performed but the catalog is still allocated and will be deleted when freed by all users.
- **DISCON** A catalog DISCONNECT has been performed but the catalog is still allocated; it will be disconnected when freed by all of its users.
- **HUNG** A hung flag has been set for this catalog.
- **-IOE** A permanent I/O error was detected for this catalog.
- **-NOM** No memory was available for catalog processing.

Address The address of the CAXWA control block

Entry The relative entry number of each CAX entry

Volume The volume on which the catalog resides
Dsname The data set name of the catalog

The following primary commands are supported directly for the CAX function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

CAX rebuild the CAX table (if the table is empty -- you can empty the table with a "XALL").

EDITT[BL] (or **ET[BL]**) enters an edit session on CAX table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:

```
F anystring [ASIS]
    [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
    [ANY/DSNAME/VOLUME]
```

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax: `LOCATE   number/dsname/volume`

O[PTIONS] provides primary command selection for CAX and operand syntax assistance.

OUT[PUT] outputs the CAX table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the CAX table based on a string match.
Syntax:

```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
           [ANY/DSNAME/VOLUME]
```

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.
Syntax:

```
SORT [NUMBER/DSNAME/VOLUME]
     [ASCEND/DESCEND]
```

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

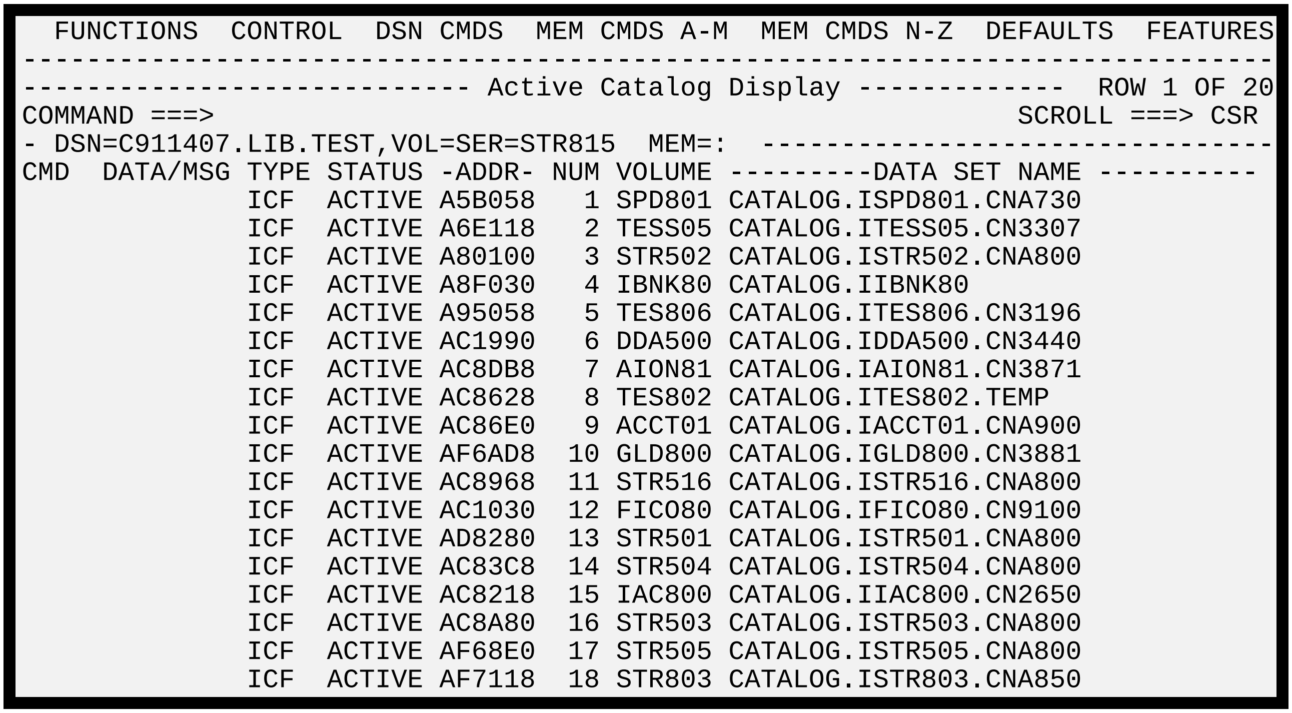
UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

X clears the CAX table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the CAX table; this is equivalent to **X ALL**

The following line commands are supported in the CAX function:
+ provide an extension panel for command entry.
= repeat the previous line command.
IDC issue an IDCAMS LISTC and direct the output to the log.
K kill and clear all following line commands.
LOG copy the line into the log.
M provide line command selection and entry assistance.
O provide line command selection and operand syntax assistance.
S select line command (normally UT, it is set by SETSEL).
TAG mark this table entry with *TAG* in the DATA/MSG field.
UT select the extended user line command panel.
X drop the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CAX function:
== = command, repeat the previous line command for the range of lines.
LL LOG command, copy the range of lines into the log.
OO O command, provide line command assistance for each line.
SS S command, select each line in the range of table lines.
XX X command, drop the range of table lines.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- Active Catalog Display -------------  ROW 1 OF 20
COMMAND ===>                                                 SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=:  ------------------------------
CMD  DATA/MSG TYPE STATUS -ADDR- NUM VOLUME ---------DATA SET NAME ----------
              ICF  ACTIVE A5B058   1 SPD801 CATALOG.ISPD801.CNA730
              ICF  ACTIVE A6E118   2 TESS05 CATALOG.ITESS05.CN3307
              ICF  ACTIVE A80100   3 STR502 CATALOG.ISTR502.CNA800
              ICF  ACTIVE A8F030   4 IBNK80 CATALOG.IIBNK80
              ICF  ACTIVE A95058   5 TES806 CATALOG.ITES806.CN3196
              ICF  ACTIVE AC1990   6 DDA500 CATALOG.IDDA500.CN3440
              ICF  ACTIVE AC8DB8   7 AION81 CATALOG.IAION81.CN3871
              ICF  ACTIVE AC8628   8 TES802 CATALOG.ITES802.TEMP
              ICF  ACTIVE AC86E0   9 ACCT01 CATALOG.IACCT01.CNA900
              ICF  ACTIVE AF6AD8  10 GLD800 CATALOG.IGLD800.CN3881
              ICF  ACTIVE AC8968  11 STR516 CATALOG.ISTR516.CNA800
              ICF  ACTIVE AC1030  12 FICO80 CATALOG.IFICO80.CN9100
              ICF  ACTIVE AD8280  13 STR501 CATALOG.ISTR501.CNA800
              ICF  ACTIVE AC83C8  14 STR504 CATALOG.ISTR504.CNA800
              ICF  ACTIVE AC8218  15 IAC800 CATALOG.IIAC800.CN2650
              ICF  ACTIVE AC8A80  16 STR503 CATALOG.ISTR503.CNA800
              ICF  ACTIVE AF68E0  17 STR505 CATALOG.ISTR505.CNA800
              ICF  ACTIVE AF7118  18 STR803 CATALOG.ISTR803.CNA850
```

Figure 9. Sample CAX Table

CHANGE Subcommand

Purpose The CHANGE subcommand switches STARWARP to a different data set.

Example CHANGE lib.cntl

Syntax

```
CHANGE  {dsn [VOLUME(vser)/VOLSET(vdef)] [ SHR/OLD] /
         * / FILE(ddname) [NUMBER(num)]}
```

Aliases STAR, START, STARTO, STARTOO, STARTOOL,
PD, PDS, PDSE, PDST, PDSTO, PDSTOO, PDSTOOL, PDSTOOLS,
C, CH, CHA, CHAN, CHANG, CHANGE

Defaults SHR or previously used data set name and volume if no operand or ***** is entered.

Required none

Operands

dsn identifies the data set name. If the data set name is not entered in quotes ('), your TSO PREFIX will be appended to the start of the entered data set name.

If ***** is entered in the data set name position, the previous set name is assumed (you are switched from the current data set to the previous data set with the ***** operand; another **CHANGE *** will switch data sets back again).

Note that if your system has password security, you may enter your data set password after the data set name and a slash (/). The syntax is: **dsn/password**

VOLUME(vser) specifies the volume name on which the data set resides. If a volume name is entered, SYSALLDA is assumed as the unit name; otherwise, the unit name from the catalog is used.

Note: this parameter should be used if the data set is not cataloged or if the catalog entry is not to be used. To use the catalog for a CHANGE subcommand even if VOLSET is in effect, you may enter **VOLUME(*)**.

VOLSET(vdef) specifies a default volume name for data set references. For example, if **VOLSET(NEWRES)** were entered, subsequent CHANGE subcommands with a data set name and no VOLUME parameter will assume a **VOLUME(NEWRES)** parameter.

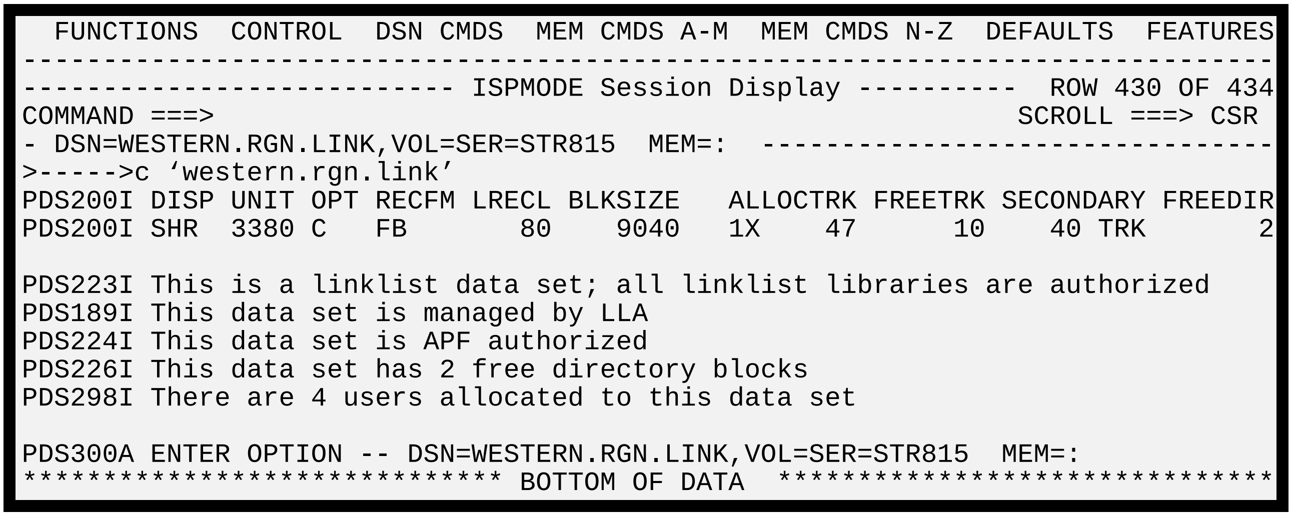
Note: this parameter should be used if a default volume is desired. To nullify the effect of VOLSET, enter **VOLSET(*)** on a subsequent CHANGE subcommand.

SHR allocate the data set with a disposition of SHR; allow simultaneous use of this data set by others. The use of **SHR** is recommended.

OLD allocate the data set with a disposition of OLD; do not allow simultaneous use of this data set by others. The use of **SHR** is recommended.

FILE(ddname) identifies the DDNAME of a preallocated data set. Note that only disk data sets (including VIO) are supported. If the FILE keyword is used, **dsn**, SHR/OLD, VOLSET and VOLUME should not also be used. However, if the data set is concatenated, the SHR or OLD keyword may be used since STARWARP reallocates the data set.

NUMBER(num) specifies the concatenation number desired for the DDNAME allocation for the FILE keyword. Note that num defaults to 1 but if num is larger than the number of concatenated data sets, the last data set in the sequence will be used.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 430 OF 434
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WESTERN.RGN.LINK,VOL=SER=STR815  MEM=:  --------------------------------
>----->c ‘western.rgn.link’
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80    9040   1X    47      10    40 TRK       2

PDS223I This is a linklist data set; all linklist libraries are authorized
PDS189I This data set is managed by LLA
PDS224I This data set is APF authorized
PDS226I This data set has 2 free directory blocks
PDS298I There are 4 users allocated to this data set

PDS300A ENTER OPTION -- DSN=WESTERN.RGN.LINK,VOL=SER=STR815  MEM=:
*****************************  BOTTOM OF DATA  *******************************
```

Figure 10. Sample CHANGE Subcommand

CMDTBL Function

Purpose The CMDTBL function manages ISPF command tables. CMDTBL can display command tables (by default, the current ISP command table is selected from the copy in memory) and manipulate command table entries. Individual command table entries can be moved, modified and tested in the CMDTBL function before saving the table.

The CMDTBL function will build, merge, or change the edit mode of the command table. After the CMDTBL function begins, you can shift left or right to see a double-line view of the command table entries.

A command table contains the specifications of general commands that can be entered from any panel during the execution of an application. Command table entries are identified by an application id and they are normally maintained in the ISPF table input library (ISPTLIB).

Example CMDTBL isp edit

Syntax

```
CMDTBL    [appl / ISP / ISR / APPLTBL / SITETBL / USERTBL ]
          [EDIT / NOEDIT                ]
          [NEW / TOP / BOTTOM / INSERT ]
          [LIB(ddname)                 ]
```

Aliases CMD, CMDT, CMDTB, CMDTBL

Defaults ISP, NOEDIT, INSERT, LIB(ISPTLIB)

Required none

Operands

appl A one to four character application name. Note that ISP, ISR, APPLTBL, SITETBL and USERTBL applications are handled as special cases and the associated table is retrieved from memory if it is active.

ISP Invoke CMDTBL with the ISP application.

ISR Invoke CMDTBL with the ISR application.

APPLTBL Invoke CMDTBL with the active application (normally, ISP or ISR).

SITETBL For ISPF 4.2 or above, invoke CMDTBL with the site-defined application.

USERTBL For ISPF 4.2 or above, invoke CMDTBL with the user-defined application.

EDIT Edit the command table.

NOEDIT Display the command table.

NEW Create a new command table.

TOP Add a command table to the top of the current command table.

BOTTOM Add a command table after the end of the current command table.

INSERT Insert a command table after the top line being displayed of the current command table.

LIB(ddname) DDNAME to use for reading the table library (this defaults to ISPTLIB).

Remarks If CMDTBL is already active, the default APPL name is the current name and no merge is done. This may be used with EDIT or NOEDIT to change the mode. For ISP, ISR, APPLTBL, SITETBL or USERTBL, LIB will be ignored and the in-memory copy will be used.

You may enter an ISPF command with any of the following methods:

- Typing the information in a COMMAND field and pressing ENTER.
- Pressing a PF key.
- Selecting an "attention" field with a light pen or cursor select key.

  After a command is entered, ISPF searches the application command table (if one exists) and then the system command table (normally ISP or ISR). If the command is found, ISPF takes the proper action immediately; otherwise, the command is passed through to the dialog without changes in the COMMAND field.

The CMDTBL function supports the following table entries:
CMD specifies line commands which affect this table entry.
DATA/MSG provides feedback on commands.
VERB specifies the name of the command table entry (this may contain 2 to 8 characters).
T (for Truncation) specifies the minimum number of characters that you must enter to find a match with the command verb (0 means that no truncation is allowed and 1 is not supported).
ACTION specifies the ISPF action to be performed when the command is entered (this may contain up to 60 characters); the following actions are supported:

> **SELECT** followed by SELECT keywords, causes a dialog (command, program or panel) to be given control. Note that this type of entry can be checked with the TEST line command. The **&ZPARM** parameter is often used in a SELECT entry; &ZPARM will be replaced by any parameters entered with the command.
> **ALIAS** followed by the name of another command and an optional parameters, specifies a command alias.
> **PASSTHRU** causes the command to be passed through to the dialog (as if it were not in the table).
> **SETVERB** causes the command to be passed through to the dialog with the command verb stored separately from the parameters.
> **NOP** causes the command to be inactivated. An "inactive command" message is displayed for this type of command.
> **blank** causes the command table entry to be ignored and scanning continues to search for additional entries with the same command verb.
> **&name** allows dynamic specification of a command table entry; any ISPF variable may be specified but a leading **&** is required.

DESCRIPTION contains a brief description of a command (it may contain up to 57 characters).

The following primary commands are supported directly for the CMDTBL function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.
APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`
CL[OSE] closes the command table application with no changes.
EDITT[BL] (or **ET[BL]**) enters an edit session on the command table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.

Syntax:
```
F anystring [ASIS]
    [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
    [ANY/VERB/ACTION/DESCRIPT]
```

IN[SERT] adds 10 blank lines to the current command table after the current row displayed at the top of the screen.

O[PTIONS] provides primary command selection for the CMDTBL function and operand syntax assistance.

OUT[PUT] outputs the command table to print or a data set.

Syntax:
```
OUTPUT [=c / F(ddname)]
```

REM[OVE] trims the command table based on a string match.

Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
        [ANY/VERB/ACTION/DESCRIPT]
```

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

SA[VE] updates the table to a data set or in memory.

Syntax:
```
SAVE appl / ISP / ISR
     [REPLACE] [NOCHECK] [LIB(ddname)]
```

Defaults: current appl, LIB(ISPTABL)

Notes: for ISP or ISR and **REPLACE**, the in-memory copy is always updated; **NOCHECK** may be used to avoid command table validity checking.

NOS[SAVE] closes the command table without making any command table updates.

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.

Syntax:
```
TAG linecmd
```

X clears the command table relative to the cursor position.

Syntax:
```
X [ABOVE/BELOW/ALL]
```

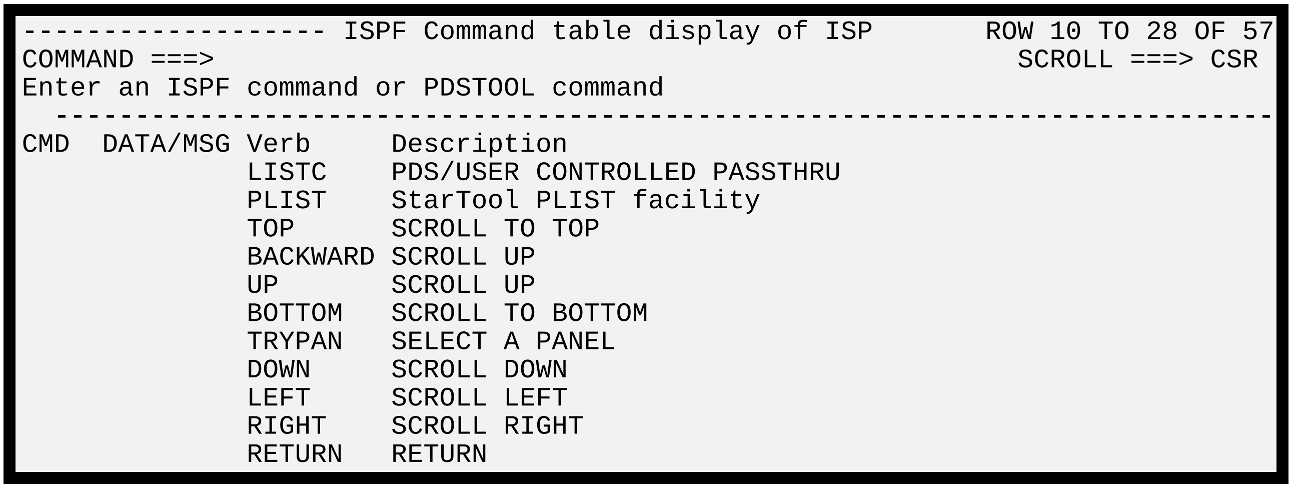
XA[LL] clears the command table; this is equivalent to **X ALL**

The following line commands are supported in the command table function:

= repeat the previous line command.
A After this entry
B Before this entry
C Copy line
DEL Delete line
In Insert line with an optional count
K Kill and clear any following line commands
LOG copy the line into the log.
M Move line
O provide line command selection assistance.
Rn Repro line with an optional count
TAG mark this table entry with *TAG* in the DATA/MSG field.
TEST Test the command table entry with prompting if required.
Xn Drop the command table entry with an optional count

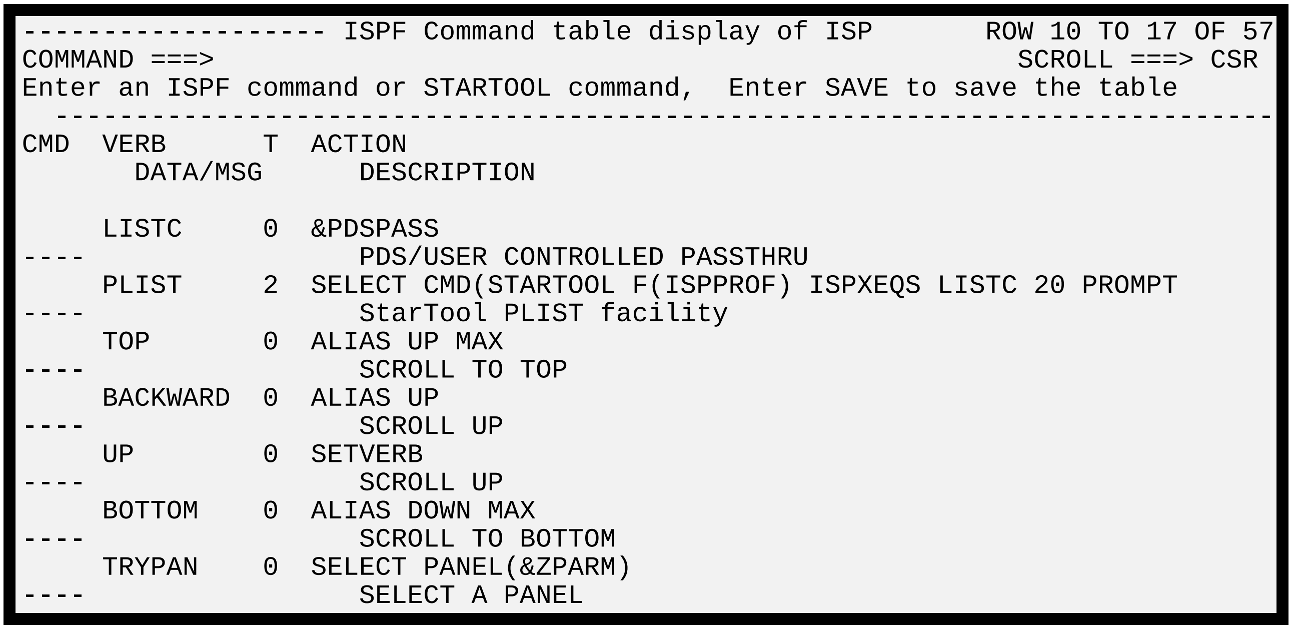
Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CSECTS function:

== = command, repeat the previous line command for the range of lines.
CC C command, copy block.
DD D command, delete block.
MM M command, move block.
RR R command, repro block.
XX X command, drop the range of table lines.



```
------------------- ISPF Command table display of ISP       ROW 10 TO 28 OF 57
COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command or PDSTOOL command
  ----------------------------------------------------------------------------
CMD  DATA/MSG Verb     Description
              LISTC    PDS/USER CONTROLLED PASSTHRU
              PLIST    StarTool PLIST facility
              TOP      SCROLL TO TOP
              BACKWARD SCROLL UP
              UP       SCROLL UP
              BOTTOM   SCROLL TO BOTTOM
              TRYPAN   SELECT A PANEL
              DOWN     SCROLL DOWN
              LEFT     SCROLL LEFT
              RIGHT    SCROLL RIGHT
              RETURN   RETURN
```

Figure 11. Sample CMDTBL Table (NOEDIT mode)



```
------------------- ISPF Command table display of ISP       ROW 10 TO 17 OF 57
COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command or STARTOOL command,  Enter SAVE to save the table
  ----------------------------------------------------------------------------
CMD  VERB      T  ACTION
       DATA/MSG      DESCRIPTION

     LISTC     0  &PDSPASS
----                 PDS/USER CONTROLLED PASSTHRU
     PLIST     2  SELECT CMD(STARTOOL F(ISPPROF) ISPXEQS LISTC 20 PROMPT
----                 StarTool PLIST facility
     TOP       0  ALIAS UP MAX
----                 SCROLL TO TOP
     BACKWARD  0  ALIAS UP
----                 SCROLL UP
     UP        0  SETVERB
----                 SCROLL UP
     BOTTOM    0  ALIAS DOWN MAX
----                 SCROLL TO BOTTOM
     TRYPAN    0  SELECT PANEL(&ZPARM)
----                 SELECT A PANEL
```

Figure 12. Sample CMDTBL Table (EDIT mode)

COMPARE Subcommand

Purpose The COMPARE subcommand displays differences between two members. The TSO COMPARE command is used; any desired COMPARE operands may be added after the member names.

If the current data set is not partitioned, an ***** may be coded in either data set name position to compare the current data set to a different data set.

If you want to compare directory entries between two libraries, use the COMPDIR subcommand (see **COMPDIR Subcommand**).

Example COMPARE   oldmem other.lib(newmem)

Syntax

```
COMPARE   oldmemb newmemb [ DELTA        ]
                          [OVSUM        ]
                          [CHNG         ]
                          [SEQ/NOSEQ    ]
                          [COBOL        ]
                          [OVOL(volname)]
                          [NVOL(volname)]

                          Note: these parameters are for SuperC.
```

Aliases CO, COM, COMP, COMPA, COMPAR, COMPARE

Defaults DELTA

Required oldmemb, newmemb

Operands

oldmemb the base member which is to be compared with newmemb. Note: oldmemb may be a data set and member name if a member from a different data set is to be used.

newmemb the new member which is to be compared with oldmemb. Note: newmemb may be a data set and member name if a member from a different data set is to be used.

If this operand is coded as the name of a partitioned data set with no member name, the base member name will be used; this can be used to compare two different levels of the same module.

other operands optional, may include any desired COMPARE operands.

Remarks Either the TSO COMPAREW (preprocessor to COMPAREX), COMPAREC (preprocessor to SuperC) or COMPARE$ (the Yale compare command) is used; this choice was made during STARWARP installation. To determine which program is used, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "COMPARE calls".

This interface is optional and should only be used if COMPAREX, SuperC or the Yale compare program is installed at your installation.

COMPDIR Subcommand

Purpose The COMPDIR subcommand creates a sublist based on differences or similarities between data sets.

Example COMPDIR pep* file(ispplib)
DEL *

Syntax

```
COMPDIR  memgroup
         {/ or dsname [VOLUME(volser)] / * / FILE(ddname) [NUMBER(numb)]}
         [ALIAS/NOALIAS                  ]
         [AS(pname) / TO(pname)          ]
               [SUBSTITUTE(oldstring/newstring)]
               [SEQ/NOSEQ]
[EXIST/NOEXIST/CHANGED/NOCHANGED/DIRCHANGE/NODIRCHANGE]
[MEMBERS/MEMLIST/ML/NEWML/SUBLIST]
```

Aliases COMPD, COMPDI, COMPDIR

Defaults memgroup, NOALIAS, EXIST, SUBLIST

Required memgroup, dsname or * or FILE(ddname)

Operands

memgroup identifies the group of members which is to be compared to those in the other data set.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** .

dsname identifies the comparison data set. Standard TSO dsname syntax should be used. Note that if ***** is entered in the dsname field, the previously used data set name used for a COMPDIR subcommand will be used.

As an alternative to the AS or TO keywords, partial member names may be entered in parentheses following the data set name as in this example: **the.dsn(pname)**

VOLUME(volser) for uncataloged data sets, identifies the output data set volume name.

FILE(ddname) specifies the output data set is already allocated with the provided ddname. Note that only disk data sets are supported.

NUMBER(numb) specifies the concatenation number desired for the DDNAME identified. Note that **numb** defaults to 1; if **numb** is larger than the number of concatenated data sets, the last data set in the concatenation sequence will be used.

ALIAS also add any associated members (aliases, apparent aliases and associated main members) to the list of names to check.

NOALIAS add only specifically identified member names to the sublist of members.

AS(pname) **pname** defines the name of any comparison members. For example, with AS(new) and the input members, AA, ABBA and IBM04, members named NEW, NEWA and NEW04 will be compared. AS and TO are equivalent keywords.

Note that if only a single member is input and the ALIAS keyword is not entered, the resulting member name is taken only from the AS or TO keyword.

TO(pname) **pname** defines the name of any comparison members. For example, with TO(new) and the input members, AA, ABBA and IBM04, members named NEW, NEWA and NEW04 will be compared. TO and AS are equivalent keywords.

Note that if only a single member is input and the ALIAS keyword is not entered, the resulting member name is taken only from the AS or TO keyword.

SUBSTITUTE(old,new) normally used with / to represent the current data set name; **SUBSTITUTE** may be used to replace a string of characters in the source data set name to create a target data set name.

For example, if the current data set is **PDSEXXX.MSGS**, the following two subcommands are equivalent:

```
COMPDIR ABC* PDSEYYYY.MSGS
COMPDIR ABC* / SUBSTITUTE(XXX/YYYY)
```

EXIST search for members from the current member group that are also present in the compare data set.

NOEXIST search for members from the current member group that are not present in the compare data set.

SEQ can be used with **CHANGED** or **NOCHANGED** to ignore sequence numbers in source member compares.

NOSEQ default, can be used with **CHANGED** or **NOCHANGED** to also compare sequence numbers in source member compares.

CHANGED search for members from the current member group that either not present in the compare data set or those members which do not contain identical data.

Note that actual member data is compared for this operand and if the members are in load data sets, you may get unequal compares if CSECTS are reblocked or reordered in load modules being compared.

NOCHANGED search for members from the current member group that are present in the compare data set and whose members contain identical data.

Note that actual member data is compared for this operand and if the members are in load data sets, you may get unequal compares if CSECTS are reblocked or reordered in load modules being compared.

DIRCHANGE search for members from the current member group that are either not present in the compare data set or those members whose directory entry is not identical to the corresponding member in the current data set.

NODIRCHANGE search for members from the current member group that are present in the compare data set and whose directory entries are identical to the corresponding member in the current data set.

MEMBERS displays the names of members which satisfy the COMPDIR search condition without changing the current member group.

MEMLIST same as **ML**. Specifies that any member selected by COMPDIR will be displayed in a MEMLIST display. If no members are selected, a null sublist is the result of the comparison.

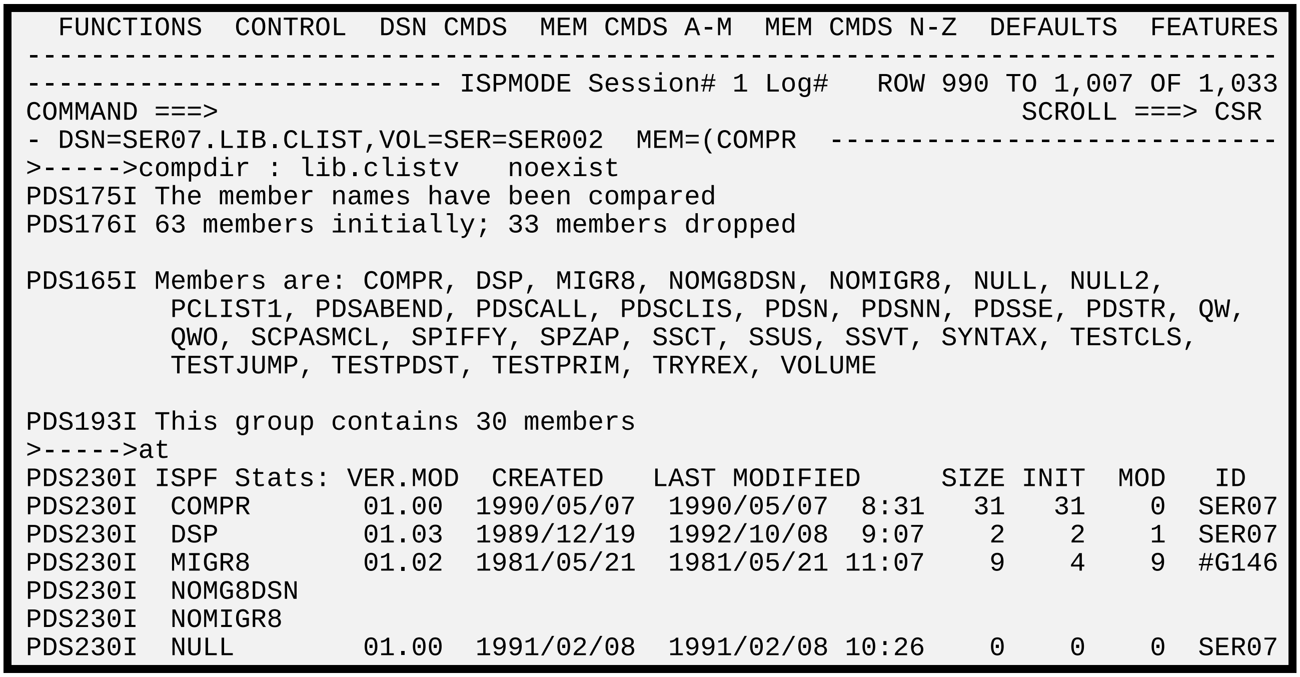
ML same as **MEMLIST**. Specifies that any member selected by COMPDIR will be displayed in a MEMLIST display. If no members are selected, a null sublist is the result of the comparison.

NEWML same as **MEMLIST** and **ML** except that the current MEMLIST is reset.

SUBLIST specifies that any members displayed by the COMPDIR subcommand will be selected for inclusion in a new SUBLIST. If no members are selected, a null sublist is the result of the comparison.

Remarks The COMPDIR subcommand is useful in comparing two libraries which contain members with a known relationship (for example, a library containing a new level of a software product or a new 'SYS1.PARMLIB' for a MVS system in test mode).

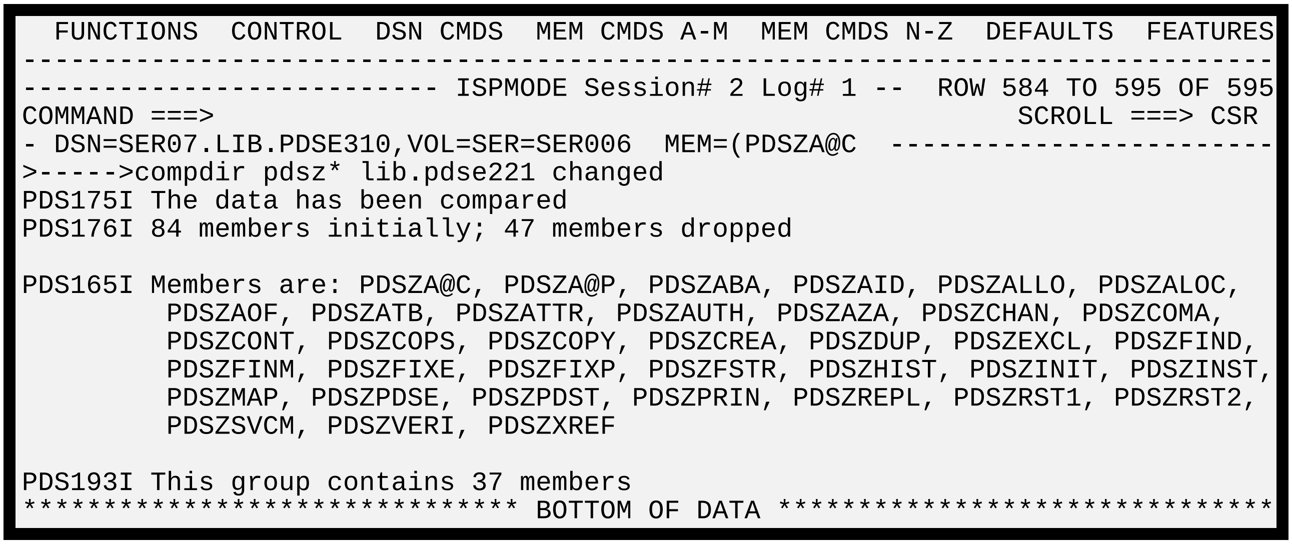
If the result of a COMPDIR subcommand is a sublist of members, you may normally follow a COMPDIR subcommand by another subcommand which performs some operation on all of the members in the sublist.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log#   ROW 990 TO 1,007 OF 1,033
COMMAND ===>                                                SCROLL ===> CSR
- DSN=SER07.LIB.CLIST,VOL=SER=SER002  MEM=(COMPR  ----------------------------
>----->compdir : lib.clistv   noexist
PDS175I The member names have been compared
PDS176I 63 members initially; 33 members dropped

PDS165I Members are: COMPR, DSP, MIGR8, NOMG8DSN, NOMIGR8, NULL, NULL2,
         PCLIST1, PDSABEND, PDSCALL, PDSCLIS, PDSN, PDSNN, PDSSE, PDSTR, QW,
         QWO, SCPASMCL, SPIFFY, SPZAP, SSCT, SSUS, SSVT, SYNTAX, TESTCLS,
         TESTJUMP, TESTPDST, TESTPRIM, TRYREX, VOLUME

PDS193I This group contains 30 members
>----->at
PDS230I ISPF Stats: VER.MOD  CREATED   LAST MODIFIED     SIZE INIT  MOD   ID
PDS230I  COMPR       01.00  1990/05/07  1990/05/07  8:31   31   31    0  SER07
PDS230I  DSP         01.03  1989/12/19  1992/10/08  9:07    2    2    1  SER07
PDS230I  MIGR8       01.02  1981/05/21  1981/05/21 11:07    9    4    9  #G146
PDS230I  NOMG8DSN
PDS230I  NOMIGR8
PDS230I  NULL        01.00  1991/02/08  1991/02/08 10:26    0    0    0  SER07
```

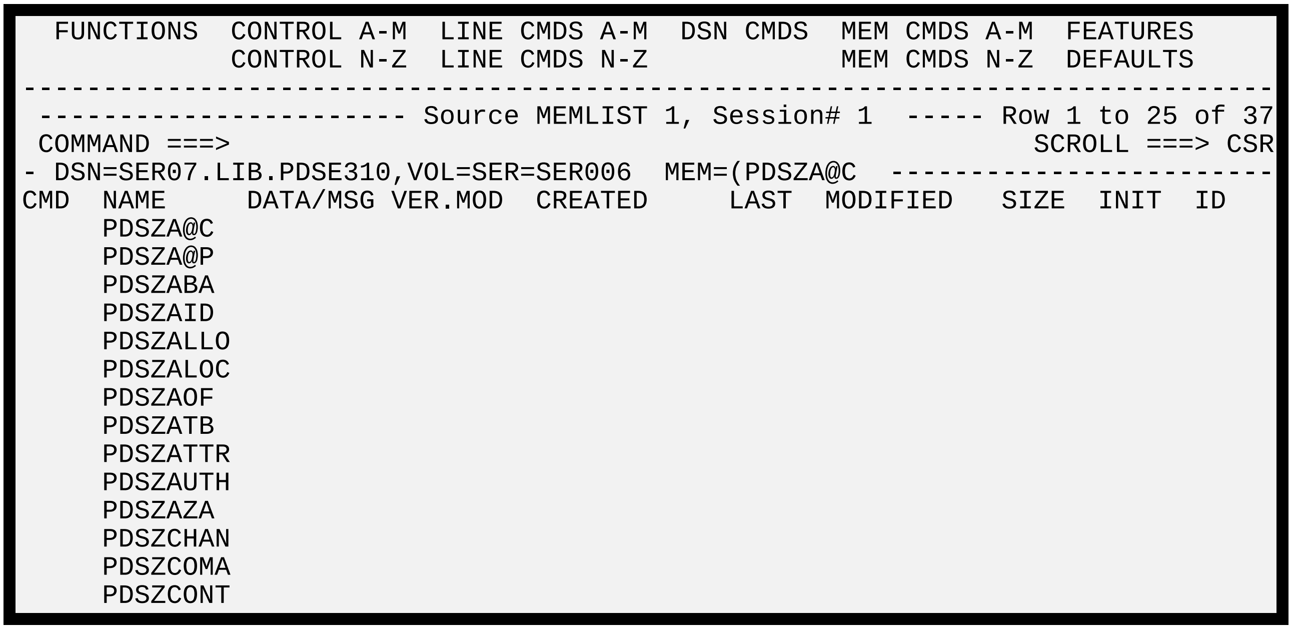
Figure 13. Sample COMPDIR Subcommand with NOEXIST

COMPDIR Subcommand



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 2 Log# 1 --  ROW 584 TO 595 OF 595
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.PDSE310,VOL=SER=SER006  MEM=(PDSZA@C  ------------------------
>----->compdir pdsz* lib.pdse221 changed
PDS175I The data has been compared
PDS176I 84 members initially; 47 members dropped

PDS165I Members are: PDSZA@C, PDSZA@P, PDSZABA, PDSZAID, PDSZALLO, PDSZALOC,
         PDSZAOF, PDSZATB, PDSZATTR, PDSZAUTH, PDSZAZA, PDSZCHAN, PDSZCOMA,
         PDSZCONT, PDSZCOPS, PDSZCOPY, PDSZCREA, PDSZDUP, PDSZEXCL, PDSZFIND,
         PDSZFINM, PDSZFIXE, PDSZFIXP, PDSZFSTR, PDSZHIST, PDSZINIT, PDSZINST,
         PDSZMAP, PDSZPDSE, PDSZPDST, PDSZPRIN, PDSZREPL, PDSZRST1, PDSZRST2,
         PDSZSVCM, PDSZVERI, PDSZXREF

PDS193I This group contains 37 members
******************************* BOTTOM OF DATA *******************************
```

Figure 17. Sample COMPDIR Subcommand with CHANGED



```
  FUNCTIONS  CONTROL A-M  LINE CMDS A-M  DSN CMDS  MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z            MEM CMDS N-Z  DEFAULTS
------------------------------------------------------------------------------
 ----------------------- Source MEMLIST 1, Session# 1  ----- Row 1 to 25 of 37
 COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.PDSE310,VOL=SER=SER006  MEM=(PDSZA@C  ------------------------
CMD  NAME     DATA/MSG VER.MOD  CREATED     LAST  MODIFIED   SIZE  INIT  ID
     PDSZA@C
     PDSZA@P
     PDSZABA
     PDSZAID
     PDSZALLO
     PDSZALOC
     PDSZAOF
     PDSZATB
     PDSZATTR
     PDSZAUTH
     PDSZAZA
     PDSZCHAN
     PDSZCOMA
     PDSZCONT
```

Figure 18. Sample Output from COMPDIR pdsz* lib.pdse221 changed memlist

CONDEND Subcommand

Purpose The CONDEND subcommand may be used to terminate STARWARP if the previous subcommand has one or more warning or error level messages. The return code will be set to the numeric value of the message that terminated STARWARP. After STARWARP is terminated by CONDEND, the return code will normally be in the range of 400 to 999; however, if RC is also entered, it will be translated to 4 or 8

Example CONDEND exit(450)

Syntax

```
CONDEND  [WARNINGS/ERRORS/EXIT(n1,n2, ...)/NOEXIT(n1,n2, ...)]
         RC
```

Aliases COND, CONDE, CONDEND

Defaults WARNINGS

Required none

Operands

WARNINGS Exit if any message issued by the last subcommand was in the range PDS400W through PDS999E (warning or error messages).

ERRORS Exit if any message issued by the last subcommand was in the range PDS600E through PDS999E (error messages).

EXIT Exit if a named warning or error message was issued by the last subcommand. For example, to exit for a PDS450W, PDS715E or PDS823E message, enter:
CONDEND EXIT(450,715,823)

NOEXIT Exit for any warning or error message except for those messages in an exemption list. For example, if you wish to exit for any warning or error message except PDS460W, PDS958E or PDS609E, enter:
CONDEND NOEXIT(460,958,609)

RC Translates the return code to four if the terminating message was between PDS400W through PDS599W or eight if the terminating message was between PDS600E and PDS999E.

Remarks This subcommand is normally used to check that a critical subcommand in a CLIST is working as expected. Normally, STARWARP CLISTS will continue to execute after program warning or error messages are received. With this facility, you can terminate a STARWARP CLIST process when it is not functioning properly and you can detect this problem from a driving CLIST by examining &LASTCC.

Placement: put CONDEND statements in your CLIST just after any critical subcommands.

CONTROL Subcommand

Purpose The CONTROL subcommand specifies global and immediate processing options.

Example CONTROL   dsname(lib.cntl(pdsdata)) old

Syntax

```
CONTROL  [ALIASINFO  / NOALIASINFO ]
         [LKEDDATE   / NOLKEDDATE  ]
         [PROMPT     / NOPROMPT    ]
         [RECOVER    / NORECOVER   ]
         [TRANSLATOR / NOTRANSLATOR]
         [SINGLE     / DOUBLE      / MULTIPLE  /  RETAIN(ntrk) ]
         [INFO       / NOINFO      ]
         [PUTLINE    / NOPUTLINE   ]
         [COMMAND    / NOCOMMAND   ]

         [SYSOUT(c)     [FORM(ffff)/NOFORM / DEST(dst)/NODEST] /
          DSNAME(memb)  [OLD/MOD/NEW/SHR]  / NODSN / NOSYSOUT ]

         [DEFAULTS    / IOSTATS    / LISTENV   /
          RESTRICTED  / TESTABEND  / TESTMSGS  /
          TESTOUTLOOP / TESTPARSE  / TESTREAD  / TESTSYNTAX ]
```

Aliases CON, CONT, CONTR, CONTRO, CONTROL

Defaults Initially in TSO:
ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR, RETAIN(9)
Initially in the background:
ALIASINFO, LKEDDATE, NOPROMPT, NORECOVER, TRANSLATOR, RETAIN(9)

Required none

Operands

ALIASINFO provide alias information for ATTRIB subcommands; also provide alias information for MAP subcommands with alias members if MAP has no CSECT-limiting parameters.

NOALIASINFO provide no alias or main member identification for ATTRIB and MAP subcommands.

LKEDDATE provide the date of the linkage edit on ATTRIB subcommands.

NOLKEDDATE do not provide the date of the linkage edit on ATTRIB subcommands.

PROMPT solicit YES/NO responses from the terminal operator unless STARWARP is executing in the background, from CLIST input or from a storage stack.

YES/NO prompting is normally performed at the following decision points:
SUBMIT before submitting a group of members

NOPROMPT do not solicit YES/NO responses; assume a YES response in each case.

RECOVER attempt ESTAE recovery after a program ABEND.

NORECOVER do not attempt ESTAE recovery after a program ABEND. Note that subcommands which specifically test for ABEND conditions (such as IF with LOADERR or VERIFY with LOAD) should recover from ABEND situations regardless of RECOVER/NORECOVER.

TRANSLATOR output assembler/compiler TRANSLATOR information from HISTORY subcommands.

NOTRANSLATOR do not output assembler/compiler TRANSLATOR information from HISTORY subcommands.

SINGLE use single buffering for input operations. This operand is ignored for PDSE data sets.

DOUBLE use double buffering for input operations wherever possible. This operand is ignored for PDSE data sets.

MULTIPLE use multiple buffering for input operations wherever possible. This mode reads an entire disk track with each input operation. This operand is ignored for PDSE data sets.

RETAIN(ntrk) specifies the number of disk track images (one through nine) to keep in storage buffers. Each new member read operation searches these buffers before performing an actual read multiple EXCP operation. Note that EXCP operations are only saved during the execution of a single subcommand. Each additional buffer requires about 58K of storage; if you later enter RETAIN(n) with a lower number, extra buffers are not returned to the system. A default number of RETAIN buffers are allocated during STARWARP initialization.

SYSOUT(c) start a session log output to the SYSOUT class specified. Note that DEST and FORM keywords can be entered as additional SYSOUT specifications. The SYSOUT output will continue until the end of the program or until CONTROL SYSOUT / NOSYSOUT / DSNAME / NODSN is entered.

NOSYSOUT terminate any SYSOUT or DSNAME session log output.

DEST(dst) specifies the destination for SYSOUT data (used only with the SYSOUT keyword). Note that dst may be entered as a 1 to 8 character JES destination or a name of the form nodename.userid

NODEST use the default destination for routing SYSOUT data.

FORM(ffff) specifies the forms name for SYSOUT (used only with the SYSOUT keyword). Note that ffff may be entered as a 1 to 4 character JES form name.

NOFORM use the default forms name for SYSOUT data.

DSNAME(memb) start a session log output to the named data set. The DSNAME output will continue until the end of the program or until CONTROL SYSOUT / NOSYSOUT / DSNAME / NODSN is entered.

In addition, note that:

1. A member name must be entered for session log output to a partitioned data set.
2. A data set password may be entered.
3. A data set disposition (OLD/MOD/SHR/NEW) may also be entered.
4. If a data set disposition of NEW is entered, the output data set will be created by the CONTROL subcommand with space parameters as follows: SPACE=(TRK,(1,4))

NODSN terminate any SYSOUT or DSNAME session log output.

OLD obtain exclusive use on the session log data set; it may not be shared with other users.

MOD add to the end of any existing session log output data set. If the data set does not exist, it will be created with DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120), SPACE=(TRK,(1,4))

SHR share the session log data set with other users. Note: since the data set is modified, the use of this parameter should be minimized.

NEW create a session log data set with DCB=(RECFM=FB, LRECL=80, BLKSIZE=3120), SPACE=(TRK,(1,4))

INFO resume output of STARWARP informational messages.

NOINFO suppress STARWARP informational messages. This mode of operation is intended for use in background sessions or in any situation where the normal informational messages are not required.

PUTLINE resume output of PUTLINE data from STARWARP.

NOPUTLINE suppress STARWARP PUTLINE output (except error messages). This mode of operation is intended for use in CLISTs where the amount of STARWARP output is to be minimized.

COMMAND resume log recording of STARWARP commands.

NOCOMMAND suppress log recording of STARWARP commands. This mode of operation is intended for use when SYSOUT/DSN recording is being performed and recording of commands and prompting messages is not desired.

DEFAULTS display a list of STARWARP defaults set by the installation default module. This list includes external programs called, global defaults and a security summary.

IOSTATS display statistics relating to input buffering. Note: only non-zero quantities are displayed and statistics are set to zero after each display.

LISTENV display information on the hardware and software environment. This information includes information on the last IPL and the system catalog.

RESTRICTED display a list of restricted use subcommand names. Subcommands can be restricted by user at STARWARP installation; this message lists subcommand and operand combinations which your installation has choosen to restrict. **Note**: this operand may be not be available for your use as it can also be restricted.

TESTABEND provide a S0C1 ABEND (or some other ABEND) for program testing.

TESTMSGS display all STARWARP numbered messages for program testing.

TESTOUTLOOP enter an intentional output message loop for program testing.

TESTPARSE switch the parse processor. When **CONTROL TESTPARSE** is first issued, the STARWARP parser is disabled in favor of IKJPARS; the next use of TESTPARSE will toggle the use of the STARWARP parser back again. When the STARWARP parser is disabled, a global message with "NOPARSE" is generated as feedback; this facility is for program testing.

TESTREAD test the input EXCP routine using an initial TTR of 000001 for each of the following cases:

- **BPAM** Read using a BPAM DCB for PDSE data sets or a BSAM DCB for striped data sets. The other following tests are not performed for these data sets. The return code should be 00 and NEXT TTR ADDRESS should be 000001.
- **SINGLE** Read single (each read obtains one physical block). The return code should be 00 and NEXT TTR ADDRESS should be 000002 if sufficient data is available.
- **DOUBLE** Read double (each read obtains two physical blocks). The return code should be 00 and NEXT TTR ADDRESS should be 000003 if sufficient data is available.
- **MULTIPLE** Read multiple (each read obtains an entire track). If the disk unit supports the READ MULTIPLE CCW, the return code should be 00 and NEXT TTR ADDRESS should be 000101 if sufficient data is available.

  If the disk unit does not support the READ MULTIPLE CCW, a PDS892E message should be received, the return code should be 00 and NEXT TTR ADDRESS should be 000003 (double buffering is used).

TESTSYNTAX test subcommand syntax only. After **CONTROL TESTSYNTAX** is first issued, STARWARP subcommands are only syntax checked (except for

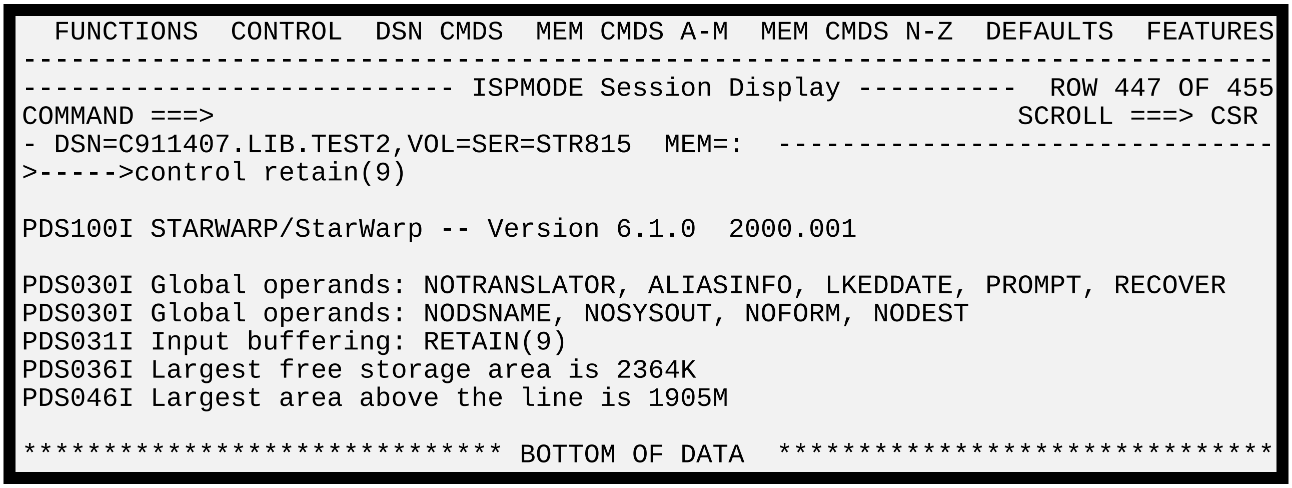
CHANGE, CONTROL, END, GO, QUIT and ISPMODE commands). The use of TESTSYNTAX will toggle the use of the STARWARP subcommands again. When the STARWARP subcommand syntax is disabled, a global message with "TESTSYNTAX" is generated as feedback; this facility may be used to test the syntax of subcommands before using them in another process.

Remarks Most CONTROL subcommand operands (such as ALIASINFO, SINGLE and TESTPARSE) are global in that they remain in effect until reset by another CONTROL subcommand. The other operands (such as TESTABEND and IOSTATS) are immediate and take effect for only a single CONTROL subcommand.

CONTROL does not utilize ISPF services; however, if STARWARP was invoked from READY mode, STARWARP attempts to establish an ISPF environment for the first CONTROL subcommand containing either a SYSOUT or DSN keyword so that subsequent subcommands can utilize ISPF services.

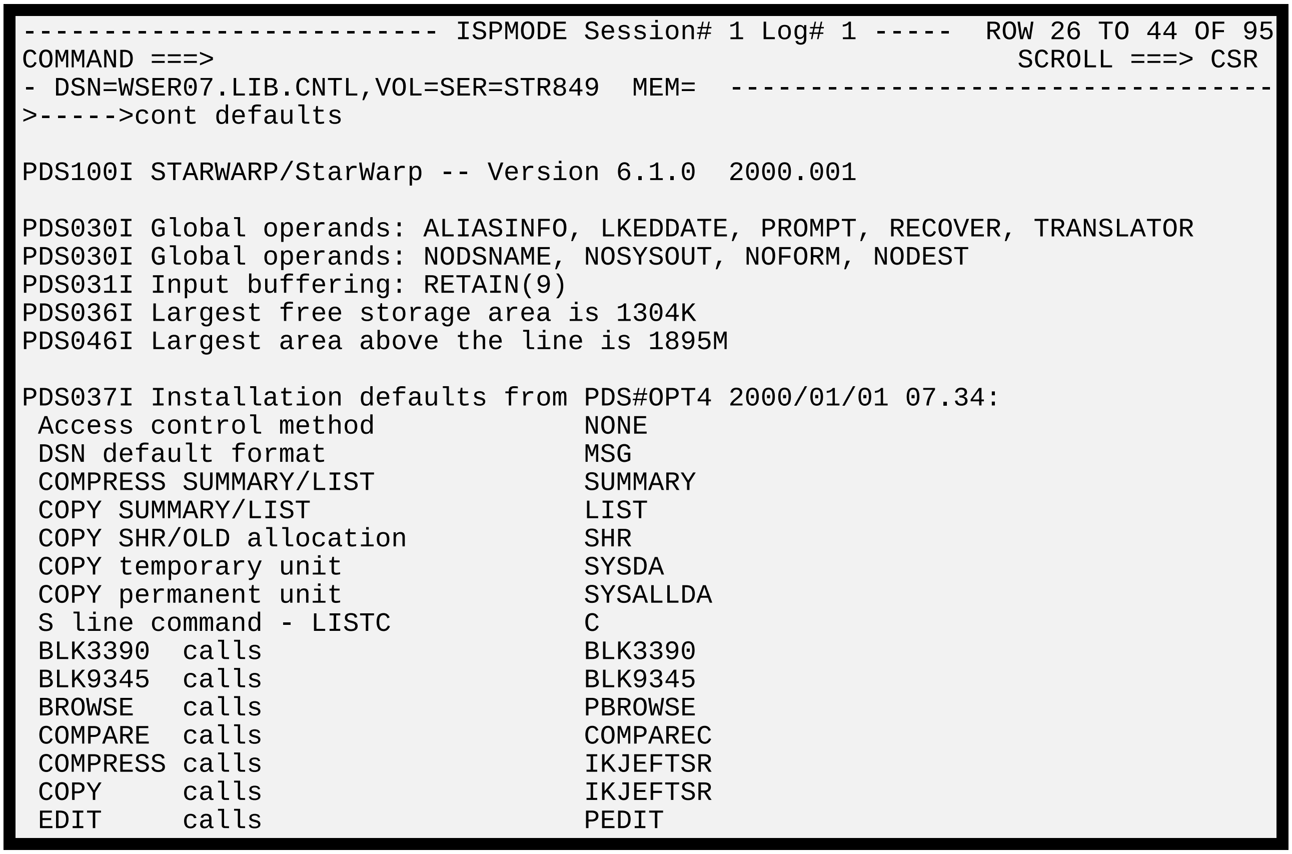
Note that the input buffering type (BPAM, SINGLE, DOUBLE, MULTIPLE or RETAIN) is reset for each data set allocated according to the data set organization or the device type on which the data set resides. RETAIN buffering is used for device types which support the READ MULTIPLE CCW and DOUBLE is used otherwise. BPAM mode is automatically selected for PDSE data sets and can not be selected explicitly.

CONTROL Subcommand



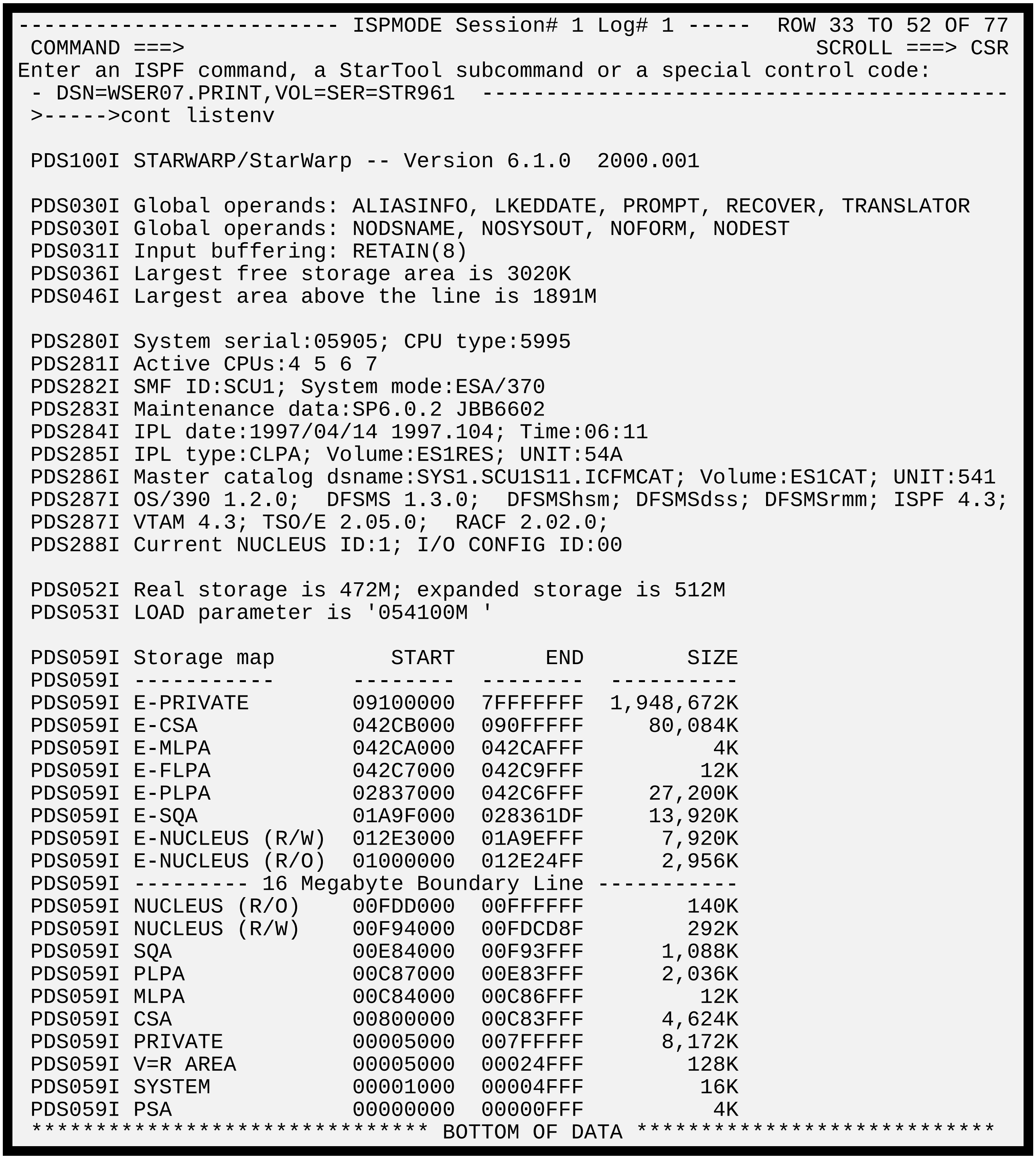
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 447 OF 455
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.TEST2,VOL=SER=STR815  MEM=:  ------------------------------
>----->control retain(9)

PDS100I STARWARP/StarWarp -- Version 6.1.0  2000.001

PDS030I Global operands: NOTRANSLATOR, ALIASINFO, LKEDDATE, PROMPT, RECOVER
PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
PDS031I Input buffering: RETAIN(9)
PDS036I Largest free storage area is 2364K
PDS046I Largest area above the line is 1905M

****************************** BOTTOM OF DATA  ******************************
```

Figure 21. Sample CONTROL Subcommand



```
------------------------- ISPMODE Session# 1 Log# 1 -----  ROW 26 TO 44 OF 95
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR849  MEM=  ---------------------------------
>----->cont defaults

PDS100I STARWARP/StarWarp -- Version 6.1.0  2000.001

PDS030I Global operands: ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR
PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
PDS031I Input buffering: RETAIN(9)
PDS036I Largest free storage area is 1304K
PDS046I Largest area above the line is 1895M

PDS037I Installation defaults from PDS#OPT4 2000/01/01 07.34:
 Access control method            NONE
 DSN default format               MSG
 COMPRESS SUMMARY/LIST            SUMMARY
 COPY SUMMARY/LIST                LIST
 COPY SHR/OLD allocation          SHR
 COPY temporary unit              SYSDA
 COPY permanent unit              SYSALLDA
 S line command - LISTC           C
 BLK3390  calls                   BLK3390
 BLK9345  calls                   BLK9345
 BROWSE   calls                   PBROWSE
 COMPARE  calls                   COMPAREC
 COMPRESS calls                   IKJEFTSR
 COPY     calls                   IKJEFTSR
 EDIT     calls                   PEDIT
```

Figure 22. Sample CONTROL DEFAULTS



```
------------------------- ISPMODE Session# 1 Log# 1 -----  ROW 33 TO 52 OF 77
 COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.PRINT,VOL=SER=STR961  -----------------------------------------
 >----->cont listenv

 PDS100I STARWARP/StarWarp -- Version 6.1.0  2000.001

 PDS030I Global operands: ALIASINFO, LKEDDATE, PROMPT, RECOVER, TRANSLATOR
 PDS030I Global operands: NODSNAME, NOSYSOUT, NOFORM, NODEST
 PDS031I Input buffering: RETAIN(8)
 PDS036I Largest free storage area is 3020K
 PDS046I Largest area above the line is 1891M

 PDS280I System serial:05905; CPU type:5995
 PDS281I Active CPUs:4 5 6 7
 PDS282I SMF ID:SCU1; System mode:ESA/370
 PDS283I Maintenance data:SP6.0.2 JBB6602
 PDS284I IPL date:1997/04/14 1997.104; Time:06:11
 PDS285I IPL type:CLPA; Volume:ES1RES; UNIT:54A
 PDS286I Master catalog dsname:SYS1.SCU1S11.ICFMCAT; Volume:ES1CAT; UNIT:541
 PDS287I OS/390 1.2.0;  DFSMS 1.3.0;  DFSMShsm; DFSMSdss; DFSMSrmm; ISPF 4.3;
 PDS287I VTAM 4.3; TSO/E 2.05.0;  RACF 2.02.0;
 PDS288I Current NUCLEUS ID:1; I/O CONFIG ID:00

 PDS052I Real storage is 472M; expanded storage is 512M
 PDS053I LOAD parameter is '054100M '

 PDS059I Storage map         START       END        SIZE
 PDS059I -----------      --------  --------  ----------
 PDS059I E-PRIVATE        09100000  7FFFFFFF  1,948,672K
 PDS059I E-CSA            042CB000  090FFFFF     80,084K
 PDS059I E-MLPA           042CA000  042CAFFF          4K
 PDS059I E-FLPA           042C7000  042C9FFF         12K
 PDS059I E-PLPA           02837000  042C6FFF     27,200K
 PDS059I E-SQA            01A9F000  028361DF     13,920K
 PDS059I E-NUCLEUS (R/W)  012E3000  01A9EFFF      7,920K
 PDS059I E-NUCLEUS (R/O)  01000000  012E24FF      2,956K
 PDS059I --------- 16 Megabyte Boundary Line -----------
 PDS059I NUCLEUS (R/O)    00FDD000  00FFFFFF        140K
 PDS059I NUCLEUS (R/W)    00F94000  00FDCD8F        292K
 PDS059I SQA              00E84000  00F93FFF      1,088K
 PDS059I PLPA             00C87000  00E83FFF      2,036K
 PDS059I MLPA             00C84000  00C86FFF         12K
 PDS059I CSA              00800000  00C83FFF      4,624K
 PDS059I PRIVATE          00005000  007FFFFF      8,172K
 PDS059I V=R AREA         00005000  00024FFF        128K
 PDS059I SYSTEM           00001000  00004FFF         16K
 PDS059I PSA              00000000  00000FFF          4K
 ******************************* BOTTOM OF DATA ****************************
```

Figure 23. Sample CONTROL LISTENV

CREATE Subcommand

Purpose The CREATE subcommand allocates a new data set based on the current data set.

Note that MODEL is also available as a full-screen front-end to CREATE as shown in an example later.

Example CREATE   new.data space(10,10)

Syntax

```
CREATE / or dataset
       [BLKSIZE(Bval)                  ]
       [DIR(num)/NODIR                 ]
       [LRECL(Lval)                    ]
       [OPTCD(W/WC/WCZ/C/CZ/Z)         ]
       [SPACE(nprim,nsec)              ]
       [BLK/TRK/CYL                    ]
       [ROUND                          ]
       [RETPD(nnnn)/EXPDT(yyddd)       ]
       [STORCLAS(sclass)               ]
       [MGMTCLAS(mclass)               ]
       [DATACLAS(dclass)               ]
       [DSNTYPE(LIBRARY/PDS)           ]
       [UNIT(Unitname)                 ]
       [VOLUME(volser)                 ]
       [SUBSTITUTE(oldstring/newstring)]
       [RECFM(F/FB/FA/FM/FBA/FBM/V/VB/VA/VM/VBA/VBM/U/UA/U)]
```

Aliases CR, CRE, CREA, CREAT, CREATE

Defaults BLKSIZE, DIR, LRECL, OPTCD, SPACE, BLK/TRK/CYL, RECFM, ROUND, EXPDT/RETPD, STORCLAS, MGMTCLAS, DATACLAS and DSNTYPE as specified for the current data set.

Required none

Operands

dataset identifies the new data set name. Standard TSO dsname syntax should be used.

BLKSIZE(Bval) specifies a DCB BLKSIZE value for the new data set. Bval defaults to its value for the current data set or may be entered as a number from 0 through 32767.

DIR(num) specifies that the number of directory blocks for the output data set. If DIR is not entered, the default is taken from the current data set.

NODIR allocate the new data set as sequential regardless of the current data set.

LRECL(Lval) specifies a DCB LRECL value for the new data set. Lval defaults to its value for the current data set or may be entered as a number from 1 through 32767.

OPTCD(Newo) specifies the desired OPTCD value for the data set. Newo defaults to its value for the current data set or may be entered as one of the following: W, WC, WCZ, C, CZ or Z.

SPACE(prim,sec) specifies the primary and secondary space allocation quantities for the new data set in block, track or cylinder units.

If BLK, TRK or CYL is not entered but SPACE is entered, the space allocation units are assumed the same as the current allocation.

BLK allocate the output data set in blocks. If BLK is entered without SPACE, the space allocation units are taken from the current allocation and converted to equivalent block units.

TRK allocate the output data set in tracks. If TRK is entered without SPACE, the space allocation units are taken from the current allocation and converted to equivalent track units.

CYL allocate the output data set in cylinder units. If CYL is entered without SPACE, the space allocation units are taken from the current allocation and converted to equivalent cylinder units.

ROUND used with BLK to specify that the data set allocation should be rounded up to cylinder boundaries.

EXPDT(yyddd) specifies the expiration date for the data set. On and after the expiration date, the data set can be deleted or written over. Note that the form EXPDT(yyyyddd) can also be used.

RETPD(nnnn) specifies the number of days (0 to 9999) that the data set should be retained by your data center.

STORCLAS(sc) specifies the storage class for the data set. This parameter is used to identify performance and availability requirements for data sets. This parameter is used instead of UNIT and VOLUME keywords for SMS managed data sets to determine where the data set is allocated. To nullify this parameter, enter STORCLAS(*).

MGMTCLAS(mc) specifies the management class for the data set. This parameter establishes the migration, backup and space release characteristics for SMS managed data sets. To nullify this parameter, enter MGMTCLAS(*).

DATACLAS(dc) specifies the data class for the data set. This parameter is used to provide data attributes such as RECFM, KEYLEN, SPACE, EXPDT or RETPD, DSORG, LRECL and some VSAM SHR options for SMS managed data sets. To nullify this parameter, enter DATACLAS(*). Note that DATACLAS can be used for data sets that are not SMS managed too.

DSNTYPE(typ) typ is specified as LIBRARY for a PDSE or PDS for a partitioned data set. A PDSE must be a SMS managed data set.

UNIT(Uname) specifies the allocation unit name. This parameter should be used if the default unit name is incorrect for this allocation.

RECFM(Newr) specifies a DCB RECFM value for the new data set. Newr defaults to its value for the current data set or may be entered as any one of the following: F, FB, FA, FBA, FM, FBM, V, VB, VA, VBA, VM, VBM, U, UA or UM.

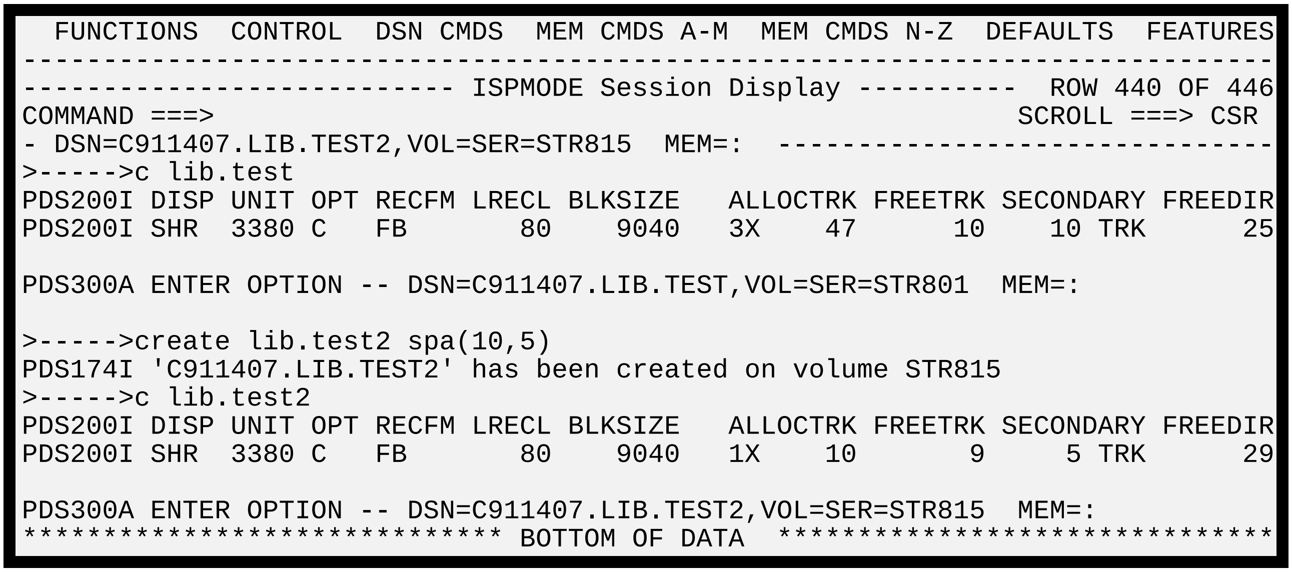
VOLUME(volser) specifies the output volume name; this parameter should be used if the data set should be allocated on a particular volume. Note: the form VOLUME(*) should be used if you wish to make another allocation on the same volume used for the current data set.

SUBSTITUTE(old,new) normally used with / to represent the current data set name; **SUBSTITUTE** may be used to replace a string of characters in the source data set name to create a target data set name.

For example, if the current data set is **PDSEXXX.MSGS**, the following two subcommands are equivalent:

```
CREATE PDSEYYYY.MSGS
CREATE / SUBSTITUTE(XXX/YYYY)
```

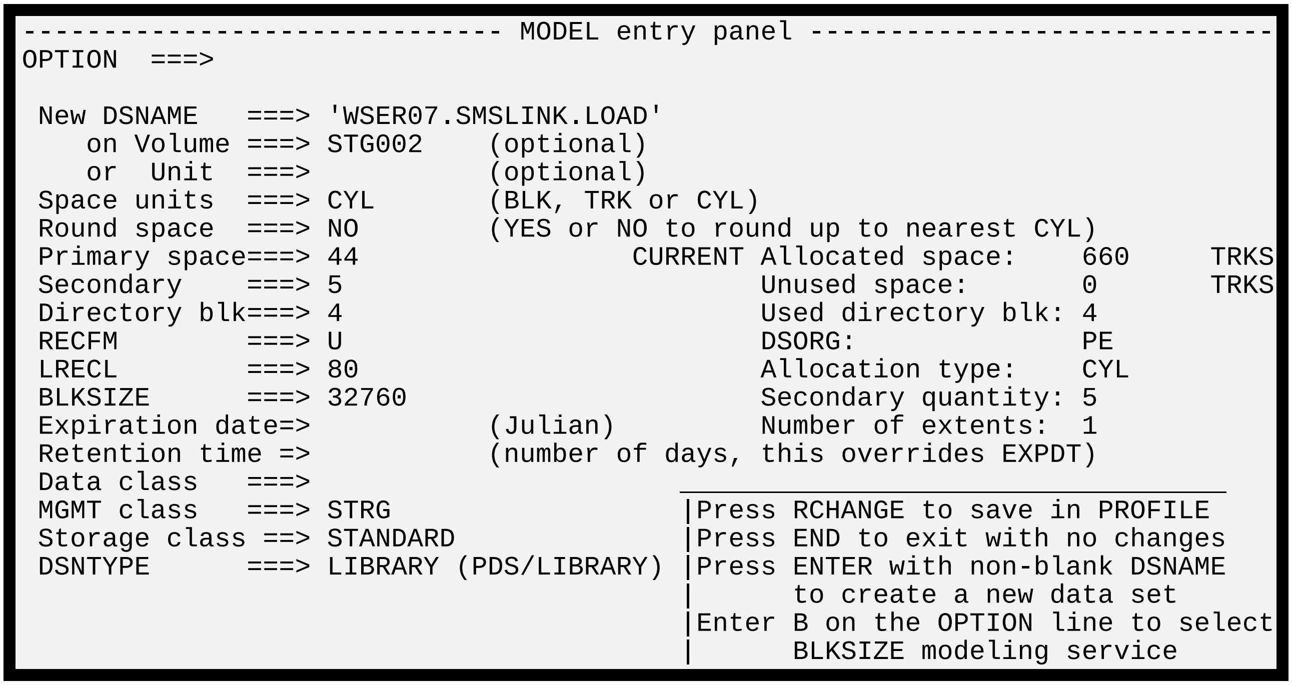
CREATE Subcommand



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 440 OF 446
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.TEST2,VOL=SER=STR815  MEM=:  ------------------------------
>----->c lib.test
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80    9040   3X    47      10    10 TRK      25

PDS300A ENTER OPTION -- DSN=C911407.LIB.TEST,VOL=SER=STR801  MEM=:

>----->create lib.test2 spa(10,5)
PDS174I 'C911407.LIB.TEST2' has been created on volume STR815
>----->c lib.test2
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80    9040   1X    10       9     5 TRK      29

PDS300A ENTER OPTION -- DSN=C911407.LIB.TEST2,VOL=SER=STR815  MEM=:
****************************** BOTTOM OF DATA  ******************************
```

Figure 26. Sample CREATE Subcommand

(Note: the parameters specified with the CREATE command will be determined by the DFSMS environment, if one exists.)



```
----------------------------- MODEL entry panel -----------------------------
OPTION  ===>

 New DSNAME   ===> 'WSER07.SMSLINK.LOAD'
    on Volume ===> STG002    (optional)
    or  Unit  ===>           (optional)
 Space units  ===> CYL       (BLK, TRK or CYL)
 Round space  ===> NO        (YES or NO to round up to nearest CYL)
 Primary space===> 44                 CURRENT Allocated space:    660     TRKS
 Secondary    ===> 5                          Unused space:       0       TRKS
 Directory blk===> 4                          Used directory blk: 4
 RECFM        ===> U                          DSORG:              PE
 LRECL        ===> 80                         Allocation type:    CYL
 BLKSIZE      ===> 32760                      Secondary quantity: 5
 Expiration date=>           (Julian)         Number of extents:  1
 Retention time =>           (number of days, this overrides EXPDT)
 Data class   ===>                     __________________________________
 MGMT class   ===> STRG                |Press RCHANGE to save in PROFILE
 Storage class ==> STANDARD            |Press END to exit with no changes
 DSNTYPE      ===> LIBRARY (PDS/LIBRARY) |Press ENTER with non-blank DSNAME
                                       |      to create a new data set
                                       |Enter B on the OPTION line to select
                                       |      BLKSIZE modeling service
```

Figure 27. Sample MODEL Panel

CSECTS Function

Purpose The CSECTS command displays a map of a load module in an ISPF table.

Example CSECTS   idcams unres

Syntax

```
CSECTS   member   [UNRESOLVED]
```

Aliases CS, CSE, CSEC, CSECT, CSECTS

Operands

member The member whose CSECTs and ENTRYs are to be mapped.
UNRESOLVED Add unresolved and weak unresolved references to the map.

Remarks

The CSECTS ISPF table is displayed in response to a CSECTS or NUCMAP command. When you are in a CSECTS display, you have many options: you may delete a part of the table, sort the table in different directions, find data in the table, print a part of the table or store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

The CSECTS table displays the names of all CSECTs and ENTRYs for a load module. You may enter line commands against CSECT table lines which will apply only to that particular CSECT. The screen fields for each line are as follows:

CMD command.   This is where a line command would be entered.
TYPE type of entry.   Any of the following may appear.

- **CSECT** CSECT entry.
- **ENTRY** ENTRY within a CSECT.
- **COMMON** COMMON area.
- **PROCEED** Private area.   This is an unnamed CSECT.
- **UNRESV** Unresolved symbol. This symbol was required but was missing when the module was last linked.
- **WEAK-U** Weak unresolved symbol. This symbol was not required and it was missing when the module was last linked.

DATA/MSG data field. Up to eight bytes of operand information can be entered here. It is used as a command feedback field -- any information in this field which is preceded by an asterisk (*) is ignored when a line command is entered. Note: this field is set to "*ENTRY" for the entry point of the module.
SECTION CSECT name.
ENTRY ENTRY name or unresolved reference name.
ADDR Address of this symbol.
LENG Hexadecimal length of this CSECT.

MD/SG Mode or segment. A segment number is provided for overlay modules; for recently linked modules, an AMODE/RMODE indication is provided as follows:

RANY RMODE of ANY and AMODE 31.
AANY RMODE of 24 and AMODE ANY.
A31 RMODE of 24 and AMODE 31.
A24 RMODE of 24 and AMODE 24.

MEM ADDR Memory address (this is for NUCMAP).

The following primary commands are supported directly for the CSECTS function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

CS[ECTS] rebuild the CSECTS table.
Syntax: `CSECTS member [UNRESOLVED]`

EDITT[BL] (or **ET[BL]**) enters an edit session on CSECT table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
      [FIRST/LAST/PREV]    [PREFIX/SUFFIX/WORD]
      [ANY/ADDRESS/NAME/ENTRY/SEGMENT/MODE]
```

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax: `LOCATE csectname/address`

O[PTIONS] provides primary command selection for the CSECTS function and operand syntax assistance.

OUT[PUT] outputs the CSECTS table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the CSECTS table based on a string match.
Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
            [ANY/ADDRESS/NAME/ENTRY/SEGMENT/MODE]
```

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.
Syntax:
```
SORT [ADDRESS/NAME]
     [ASCEND/DESCEND]
```

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation. The CSECTS UT panel (actually the same panel as the log) allows dynamic primary commands with which you specify command names and their corresponding actions.

X clears the CSECTS table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the CSECTS table; this is equivalent to **X ALL**

The following line commands are supported in the CSECTS function:

+ provide an extension panel for command entry.

= repeat the previous line command.

BASE assign a base value at this CSECT for a ZAP line command on a later CSECT.

DELI DELINK the CSECT.

DIS DISASM the CSECT.

F FIND data in the CSECT.

H HISTORY for the CSECT.

K kill and clear all following line commands.

L LIST data from the CSECT.

LOG copy the line into the log.

M provide line command selection and entry assistance.

O provide line command selection and operand syntax assistance.

R REPLACE data in the CSECT.

S select line command (normally LIST, it is set by SETSEL).

TAG mark this table entry with *TAG* in the DATA/MSG field.

UT select the extended user line command panel.

X drop the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CSECTS function:

== = command, repeat the previous line command for the range of lines.

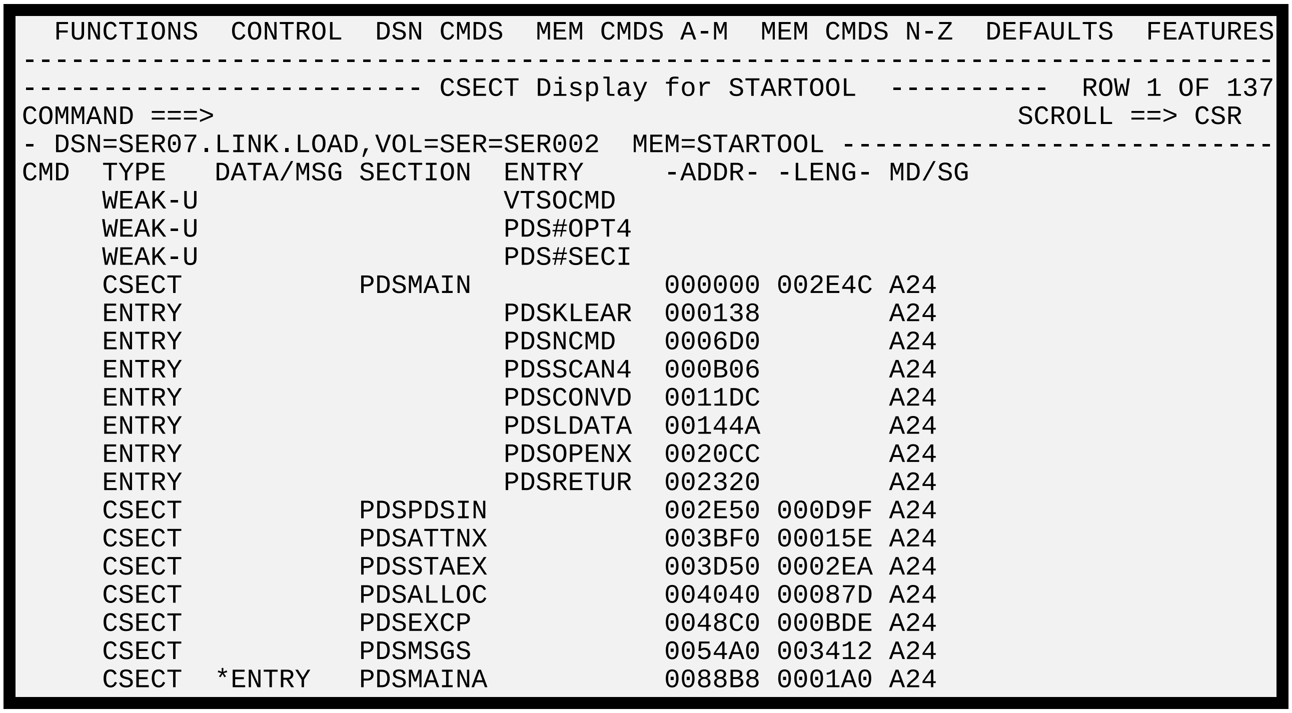
LL LIST command, list data from each CSECT.

OO O command, provide line command assistance for each line.

SS S command, select each line in the range of table lines.

XX X command, drop the range of table lines.

CSECTS Function



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- CSECT Display for STARTOOL  ----------  ROW 1 OF 137
COMMAND ===>                                                  SCROLL ==> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=STARTOOL ---------------------------
CMD  TYPE   DATA/MSG SECTION  ENTRY     -ADDR- -LENG- MD/SG
     WEAK-U                   VTSOCMD
     WEAK-U                   PDS#OPT4
     WEAK-U                   PDS#SECI
     CSECT           PDSMAIN            000000 002E4C A24
     ENTRY                    PDSKLEAR  000138        A24
     ENTRY                    PDSNCMD   0006D0        A24
     ENTRY                    PDSSCAN4  000B06        A24
     ENTRY                    PDSCONVD  0011DC        A24
     ENTRY                    PDSLDATA  00144A        A24
     ENTRY                    PDSOPENX  0020CC        A24
     ENTRY                    PDSRETUR  002320        A24
     CSECT           PDSPDSIN           002E50 000D9F A24
     CSECT           PDSATTNX           003BF0 00015E A24
     CSECT           PDSSTAEX           003D50 0002EA A24
     CSECT           PDSALLOC           004040 00087D A24
     CSECT           PDSEXCP            0048C0 000BDE A24
     CSECT           PDSMSGS            0054A0 003412 A24
     CSECT  *ENTRY   PDSMAINA           0088B8 0001A0 A24
```

Figure 28. Sample CSECTS Table

DCF Subcommand

Purpose The DCF subcommand prints a hardcopy list of a member. The TSO SCRIPT command is used; any desired SCRIPT operands may be added after the member name.

Example DCF mema:memb twopass index file(myfile)

Syntax

```
DCF  memgroup
     BIND(obind {ebind})           PAGE{({PROMPT}{{FROM} p {TO} q}
     CHARS(font1..font4)           {{FROM} p FOR n}{{FROM} p ON})}
     CONTINUE/NOCONT               PRINT{(copies,class,fcb,ucs)}
     CTF                           PROFILE{(fileid)}/NOPROFILE
     DDUT/NODDUT                   QUIET
     DEST(station-id)              SEARCH(libname)(opnum...)
     DEVICE(devtype)               SEGLIB/NOSEGLIB
     FILE{(fileid)}                SPELLCHK
     FONTLIB({ftype}{fmodel})      STOP
     FPASSES n                     SYON/SYOFF
     INDEX                         SYSVAR(n value...)
     LIB(libename...)(opnum...)    TERM
     MESSAGE({DELAY}{D}{TRACE})    TLIB
     NOSPIE                        TWOPASS
     NOWAIT                        UNFORMAT
     NUMBER                        UPCASE
     OPTIONS{(fileid)}             @user-option
```

Aliases DC, DCF

Defaults memgroup

Required none

Operands

memgroup identifies the members(s) to be printed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

others optional, may include any desired TSO SCRIPT operands.

Remarks This interface is optional and should only be used if your installation has DCF Script installed.

DDNAME Function

Purpose

The DDNAME command is used to display a subset of your TSO session allocations in the LISTA table. The DDNAME/LISTA table is maintained in DDNAME and concatenation order (the SORT command is not supported).

Example

DDNAME isp*

Syntax

```
DDNAME namemask
```

Aliases

DD, DDN, DDNA, DDNAM, DDNAME

Defaults

none

Required

namemask

Operands

namemask this is a partial DDNAME which may contain masking characters (***** for combinations, **/** for patterns and **:** for ranges). For example, **ISP*LIB**, **PROC/SYS** and **ISP:ISR**.

Remarks

The DDNAME/LISTA ISPF table is displayed in response to a DDNAME command. When you are in a DDNAME/LISTA display, you have many options: you may delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

DDNAME has alternate panels, as do most of the ISPMODE tables in STARWARP, containing extra information about the listed data sets. The first DDNAME/LISTA panel shows basic allocation information such as DDNAME, data set TYPE, Open count (under the header O#), VOLUME allocated, MEMBER name and DSNAME. Command error messages are provided under the DATA/MSG header as follows:

***INVALID**	this line command is not supported.
***DD GONE**	this file is not allocated.
***OPEN DD**	this file is open.
***IN USE**	this file is in-use.
***NOT CON**	this file is not concatenated.
***PERMCON**	this file is permanently concatenated.
NO DDN	this file has a blank ddname.

TYPE will be the data set DSORG or "VIO", "CTLG", "VVDS", "JES" or "TERM" depending on the type of allocation.

The second DDNAME/LISTA panel is a double line panel which includes some additional information: DISP for data set disposition and STAT for data set status. The STAT field will be **TMP** for temporary, **PRM** for permanent or **CNV** for convertible.

The following primary commands are supported directly for the LISTA/DDNAME function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

ALT[ERNAT] displays an alternate view of the LISTA/DDNAME table.

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

BAT[CHJCL] builds batch JCL which will include all DD statements in the DDNAME table.

DUA[L] displays a double line view of the LISTA/DDNAME table.

DD[NAME] rebuilds the LISTA/DDNAME table selecting data sets with a DDNAME mask.
Syntax: `DDNAME ddnamemask`

EDITT[BL] (or **ET[BL]**) enters an edit session on LISTA/DDNAME table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
  [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
  [ANY/DDNAME/DSNAME/VOLUME/MEMBER/ALLOC]
```

FI[ND] global command, changes to each data set in the table and issues a FIND subcommand. Since the syntax entered on a FIND subcommand is actually applied to each data set individually, you should not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.
Syntax:
```
FIND memgroup 'anystring'
     [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
     [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
```

GL[OBAL] global command, changes to each data set in the table and issues a STARWARP subcommand.
Syntax: `GLOBAL anysubcommand [anyoperands]`

LE[FT] rotates through alternate views of the LISTA/DDNAME table. PF keys 10 and 22 are normally set to LEFT.

LISTA rebuilds the LISTA/DDNAME table selecting data sets by attributes.
Syntax: `LISTA [attributes]`

MODEL global command, changes to each data set in the table and issues a MODEL command.

NOR[MAL] displays the default view of the LISTA/DDNAME table.

OUT[PUT] outputs the LISTA/DDNAME table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the LISTA/DDNAME table based on a string match.
Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
       [ANY/DDNAME/DSNAME/VOLUME/MEMBER/ALLOC]
```

REPL[ACE] global command, changes to each data set in the table and issues a REPLACE subcommand. Since the syntax entered on a REPLACE subcommand is actually applied to each data set individually, you should not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.
Syntax:
```
REPLACE memgroup 'fromstring' 'tostring'
        [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
        [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
        [WRITE/NOWRITE]
```

RESET often used with data set tagging, clears the DATA/MSG field in all table entries.

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the LISTA/DDNAME table. PF keys 11 and 23 are normally set to RIGHT.

SEEK global command, changes to each data set in the table and looks for a member.
Syntax: `SEEK member`

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: TAG linecmd

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

X clears the LISTA/DDNAME table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

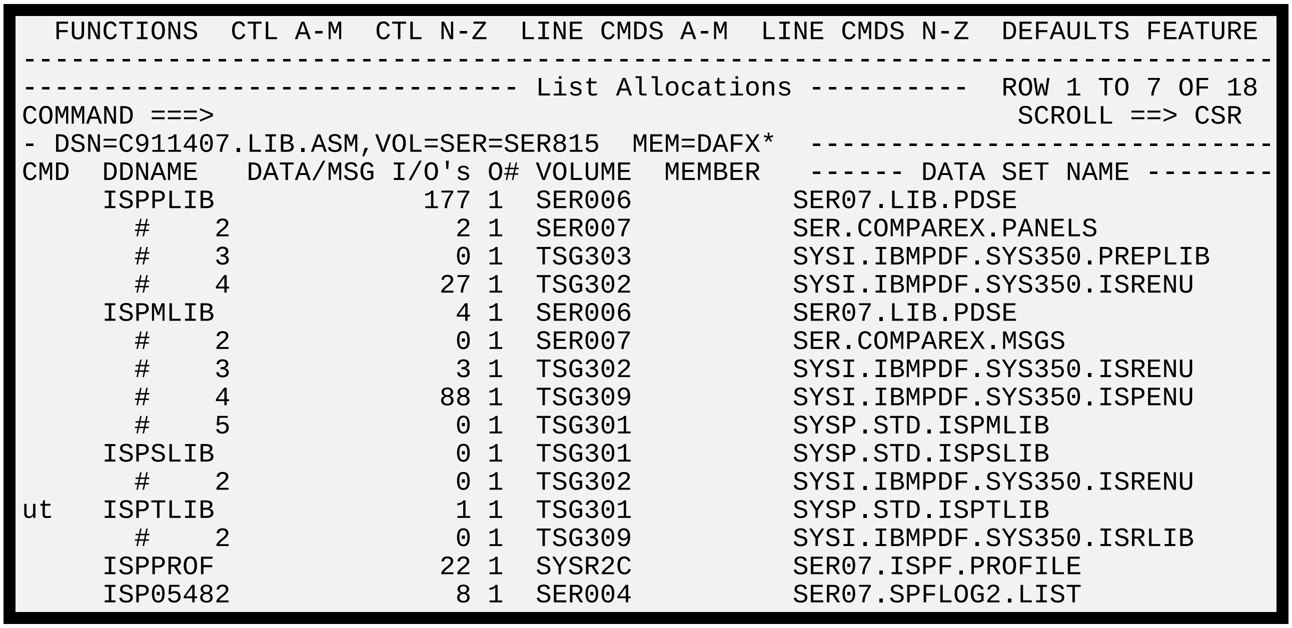
XA[LL] clears the LISTA/DDNAME table; this is equivalent to **X ALL**

The following line commands are supported in the LISTA/DDNAME function:

+ provide an extension panel for command entry.
= repeat the previous line command.
6 change to the data set and MEMLIST all.
A change to the data set and execute an alternate subcommand.
B change to the data set and MEMLIST all.
C change to the data set.
DCAT deconcatenate non-permanent data sets in a concatenated group.
E change to the data set and MEMLIST all.
FREE free a DDNAME or all data sets in a concatenated group.
GO change to the data set using GO processing (a number is optional).
IDC perform an IDCAMS LISTC and direct output to the log.
K kill and clear all following line commands.
LC add the data set name to the current LISTC/LISTF table.
LOG copy the line into the log.
M provide line command selection and entry assistance.
ML change to the data set and MEMLIST all.
MOD change to the data set and MODEL.
O provide line command selection and operand syntax assistance.
S select line command (normally CHANGE, it is set by SETSEL).
SEEK change to the data set and check for a member.
SETA save a STARWARP subcommand for repeated use by the ALT line command.
TAG mark this table entry with *TAG* in the DATA/MSG field.
U change to the data set and display USAGE.
UT select the extended user line command panel.
W add the data set name (and any member name) to the WORKPAD.
WHO check for users of this data set with the WHOHAS subcommand.
X drop the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTA/DDNAME function:

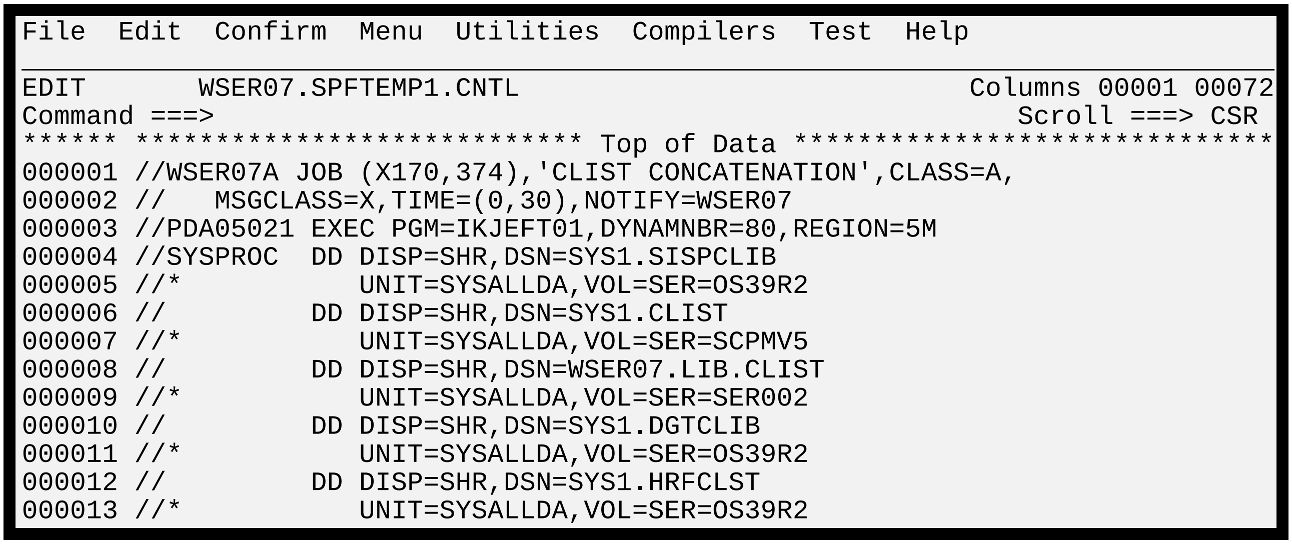
== = command, repeat the previous line command for the range of lines.
AA A command, change and execute an alternate subcommand for each line.
BB B command, change and MEMLIST all for each line.
CC C command, change data set for each line.
EE E command, change and MEMLIST all for each line.
GG GO command, change to each data set using GO processing.
LL LOG command, copy the range of lines into the log.
MM M command, change and MEMLIST all for each line.
OO O command, provide line command assistance for each line.
SS S command, select each line in the range of table lines.
SSEE SEEK command, change to each data set and check for a member.
UU U command, change and display USAGE for each line.
XX X command, drop the range of table lines.



```
  FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS FEATURE
-------------------------------------------------------------------------------
------------------------------ List Allocations ----------  ROW 1 TO 7 OF 18
COMMAND ===>                                                  SCROLL ==> CSR
- DSN=C911407.LIB.ASM,VOL=SER=SER815  MEM=DAFX*  -----------------------------
CMD  DDNAME   DATA/MSG I/O's O# VOLUME  MEMBER   ------ DATA SET NAME --------
     ISPPLIB             177 1  SER006          SER07.LIB.PDSE
       #    2              2 1  SER007          SER.COMPAREX.PANELS
       #    3              0 1  TSG303          SYSI.IBMPDF.SYS350.PREPLIB
       #    4             27 1  TSG302          SYSI.IBMPDF.SYS350.ISRENU
     ISPMLIB               4 1  SER006          SER07.LIB.PDSE
       #    2              0 1  SER007          SER.COMPAREX.MSGS
       #    3              3 1  TSG302          SYSI.IBMPDF.SYS350.ISRENU
       #    4             88 1  TSG309          SYSI.IBMPDF.SYS350.ISPENU
       #    5              0 1  TSG301          SYSP.STD.ISPMLIB
     ISPSLIB               0 1  TSG301          SYSP.STD.ISPSLIB
       #    2              0 1  TSG302          SYSI.IBMPDF.SYS350.ISRENU
ut   ISPTLIB               1 1  TSG301          SYSP.STD.ISPTLIB
       #    2              0 1  TSG309          SYSI.IBMPDF.SYS350.ISRLIB
     ISPPROF              22 1  SYSR2C          SER07.ISPF.PROFILE
     ISP05482              8 1  SER004          SER07.SPFLOG2.LIST
```

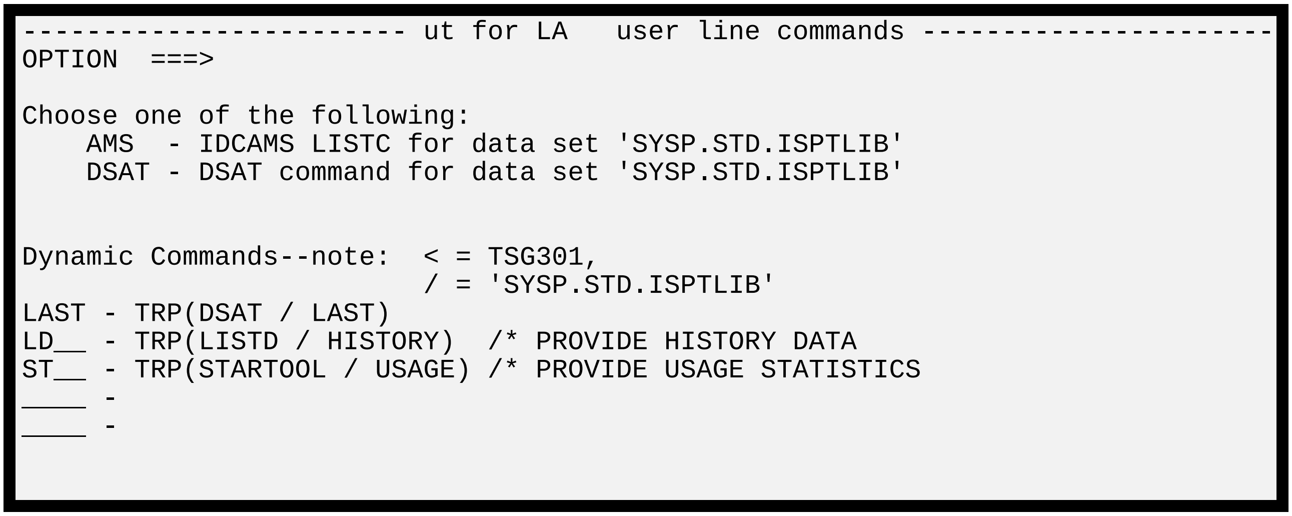
Figure 29. Sample DDNAME Table

DDNAME Function



```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
______________________________________________________________________________
EDIT       WSER07.SPFTEMP1.CNTL                            Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //WSER07A JOB (X170,374),'CLIST CONCATENATION',CLASS=A,
000002 //   MSGCLASS=X,TIME=(0,30),NOTIFY=WSER07
000003 //PDA05021 EXEC PGM=IKJEFT01,DYNAMNBR=80,REGION=5M
000004 //SYSPROC  DD DISP=SHR,DSN=SYS1.SISPCLIB
000005 //*           UNIT=SYSALLDA,VOL=SER=OS39R2
000006 //         DD DISP=SHR,DSN=SYS1.CLIST
000007 //*           UNIT=SYSALLDA,VOL=SER=SCPMV5
000008 //         DD DISP=SHR,DSN=WSER07.LIB.CLIST
000009 //*           UNIT=SYSALLDA,VOL=SER=SER002
000010 //         DD DISP=SHR,DSN=SYS1.DGTCLIB
000011 //*           UNIT=SYSALLDA,VOL=SER=OS39R2
000012 //         DD DISP=SHR,DSN=SYS1.HRFCLST
000013 //*           UNIT=SYSALLDA,VOL=SER=OS39R2
```

Figure 30. Sample DDNAME Output from BATCHJCL



```
------------------------ ut for LA   user line commands ----------------------
OPTION  ===>

Choose one of the following:
    AMS  - IDCAMS LISTC for data set 'SYSP.STD.ISPTLIB'
    DSAT - DSAT command for data set 'SYSP.STD.ISPTLIB'


Dynamic Commands--note:  < = TSG301,
                         / = 'SYSP.STD.ISPTLIB'
LAST - TRP(DSAT / LAST)
LD__ - TRP(LISTD / HISTORY)  /* PROVIDE HISTORY DATA
ST__ - TRP(STARTOOL / USAGE) /* PROVIDE USAGE STATISTICS
____ -
____ -
```

Figure 31. Sample DDNAME user line command

DECODE Subcommand

Purpose The DECODE subcommand decrypts a member. This was written using the data encryption algorithm submitted by IBM to the National Bureau of Standards and published in the Federal Register.

Example DECODE   mymember plaintxt keyexamp

Syntax

```
DECODE input output key [SHR/OLD    ]
```

Aliases DEC, DECO, DECOD, DECODE

Defaults OLD

Required input, output, key

Operands

input identifies the member to be decoded.

If this is a 1-8 byte simple name it is taken to be a member name in the current data set; otherwise it is taken to be data.set(member).

output identifies the receiving member.

If this is a 1-8 byte simple name it is taken to be a member name in the current data set; otherwise it is taken to be data.set(member).

key This is a 1-8 byte alphameric key which is to be used for decoding the data set member.

OLD This allocation is not to be shared with any other allocation for this data set.

SHR This allocation is to be shared with other allocations for this data set.

Remarks This subcommand decrypts data using the supplied translation key. Since this algorithm works on eight bytes at a time, the actual number of bytes enciphered or deciphered will be an integral number of eight bytes that is less than or equal to the record length. Note that the greatest number of bytes left unchanged on a record would be the rightmost seven bytes.

Warning: this method does not record your key. If you cannot provide the key, the data will not be readable. Note that the DECODE load module may be used independently of STARWARP. A HELP entry is available but the basic syntax is as follows:

DECODE input.dsn(memin) output.dsn(memout) key SHR/OLD

DELINK Subcommand

Purpose The DELINK subcommand reconstructs object code from a load module.

Example DELINK mymemb

Syntax

```
DELINK   memgroup [MODULE({* / Fullm / Partm*)} ]
```

Aliases DELI, DELIN, DELINK

Defaults memgroup

Required none

Operands

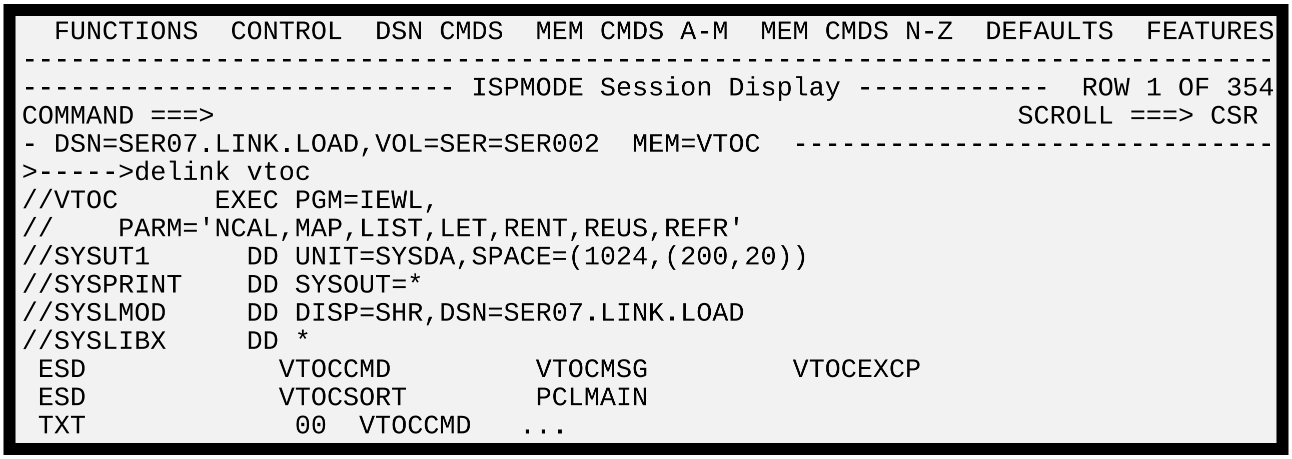
memgroup identifies the member(s) to be decoded to object code.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT or ENTRY names to be considered.

The MODULE operand has several valid forms:

MODULE(*)	use the previous name entered on any MODULE keyword.
MODULE(Fullm)	decode only a CSECT or ENTRY named FULLM.
MODULE(Partm*)	decode only a CSECT or ENTRY named PARTM...



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ------------  ROW 1 OF 354
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=VTOC  -----------------------------
>----->delink vtoc
//VTOC      EXEC PGM=IEWL,
//    PARM='NCAL,MAP,LIST,LET,RENT,REUS,REFR'
//SYSUT1      DD UNIT=SYSDA,SPACE=(1024,(200,20))
//SYSPRINT    DD SYSOUT=*
//SYSLMOD     DD DISP=SHR,DSN=SER07.LINK.LOAD
//SYSLIBX     DD *
 ESD            VTOCCMD         VTOCMSG         VTOCEXCP
 ESD            VTOCSORT        PCLMAIN
 TXT             00  VTOCCMD   ...
```

Figure 33. Sample DELINK Subcommand

DIRENTRY Subcommand

Purpose The DIRENTRY subcommand displays a member's directory entry.

Example DIRENTRY   mema:memb

Syntax

```
DIRENTRY   memgroup [SHORT/LONG]
```

Aliases DIR, DIRE, DIREN, DIRENT, DIRENTR, DIRENTRY

Defaults memgroup, SHORT for THEN(DIR) or ELSE(DIR); otherwise, LONG

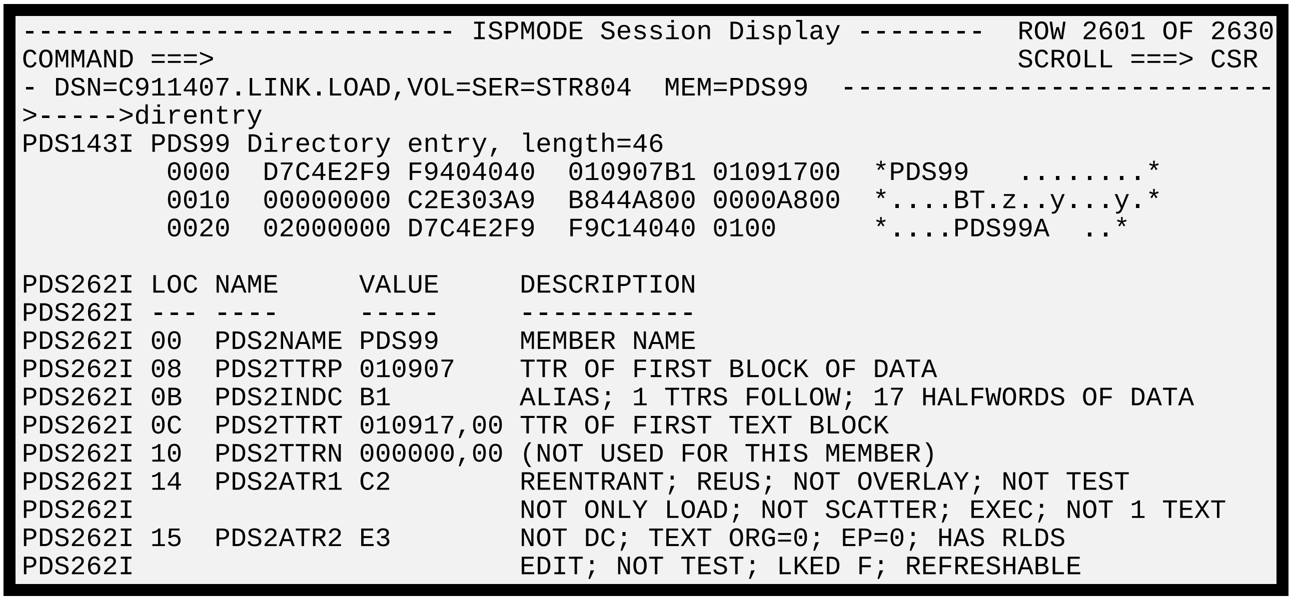
Required none

Operands

memgroup identifies the member(s) whose directory entry is to be displayed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

SHORT provide only a directory entry dump.
LONG provide a directory entry dump and a display of individual elements from the directory entry.



```
--------------------------- ISPMODE Session Display --------  ROW 2601 OF 2630
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=PDS99 --------------------------
>----->direntry
PDS143I PDS99 Directory entry, length=46
         0000  D7C4E2F9 F9404040  010907B1 01091700  *PDS99   ........*
         0010  00000000 C2E303A9  B844A800 0000A800  *....BT.z..y...y.*
         0020  02000000 D7C4E2F9  F9C14040 0100      *....PDS99A  ..*

PDS262I LOC NAME     VALUE     DESCRIPTION
PDS262I --- ----     -----     -----------
PDS262I 00  PDS2NAME PDS99     MEMBER NAME
PDS262I 08  PDS2TTRP 010907    TTR OF FIRST BLOCK OF DATA
PDS262I 0B  PDS2INDC B1        ALIAS; 1 TTRS FOLLOW; 17 HALFWORDS OF DATA
PDS262I 0C  PDS2TTRT 010917,00 TTR OF FIRST TEXT BLOCK
PDS262I 10  PDS2TTRN 000000,00 (NOT USED FOR THIS MEMBER)
PDS262I 14  PDS2ATR1 C2        REENTRANT; REUS; NOT OVERLAY; NOT TEST
PDS262I                        NOT ONLY LOAD; NOT SCATTER; EXEC; NOT 1 TEXT
PDS262I 15  PDS2ATR2 E3        NOT DC; TEXT ORG=0; EP=0; HAS RLDS
PDS262I                        EDIT; NOT TEST; LKED F; REFRESHABLE
```

Figure 34. Sample DIRENTRY Subcommand

DISASM Subcommand

Purpose The DISASM subcommand reconstructs assembler language instructions from a module; this procedure is often referred to as "decoding" or "disassembling" instructions.

Example DISASM   mymemb offset(1af2)

Syntax

```
DISASM   memgroup [FLOAT/NOFLOAT                ]
                  [MODULE({* / Fullm / Partm*)} ]
                  [OFFSET(Hx)                   ]
                  [MVSXA/ESA370                 ]
                  [PRIV/NOPRIV                  ]
                  [REASM                        ]
```

Aliases DISA, DISAS, DISASM

Defaults memgroup, NOFLOAT, NOPRIV, ESA370

Required none

Operands

memgroup identifies the member(s) to be disassembled.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

FLOAT include floating-point instructions in the instruction set.
NOFLOAT do not decode floating-point instructions.
MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT or ENTRY names to be displayed.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected. The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.
MODULE(Fullm) format only a CSECT or ENTRY named FULLM.
MODULE(Partm*) format only a CSECT or ENTRY named PARTM...

OFFSET(Hx) specifies a 1 to 6 digit hexadecimal module offset at which the display is to begin. If both MODULE and OFFSET keywords are entered, the offset applies to each module selected.
MVSXA use only instructions defined on MVS/XA systems.
ESA370 use only instructions defined on MVS/ESA systems; this instruction set also includes all MVS/XA instructions.
PRIV include privileged instructions in the instruction set.
NOPRIV do not decode privileged instructions.

REASM output the disassembled instructions in a format which can be submitted to recreate the decoded module. When this format is requested, the module offset field and the CSECT offset field (the first two fields of output) are reversed in order and placed at the end of each decoded statement.

The normal PDS141I header messages are suppressed and JCL is created to assemble and link the decoded module and its aliases.

Remarks DISASM formats its output as follows:

module offset a hexadecimal offset in the current module. Note: with REASM format, this would be the last output field.

CSECT offset a hexadecimal offset in the current CSECT. Note: with REASM format, this field would be just before the last output field.

name a external name or generated internal name (format: "A" followed by a six character offset in the module).

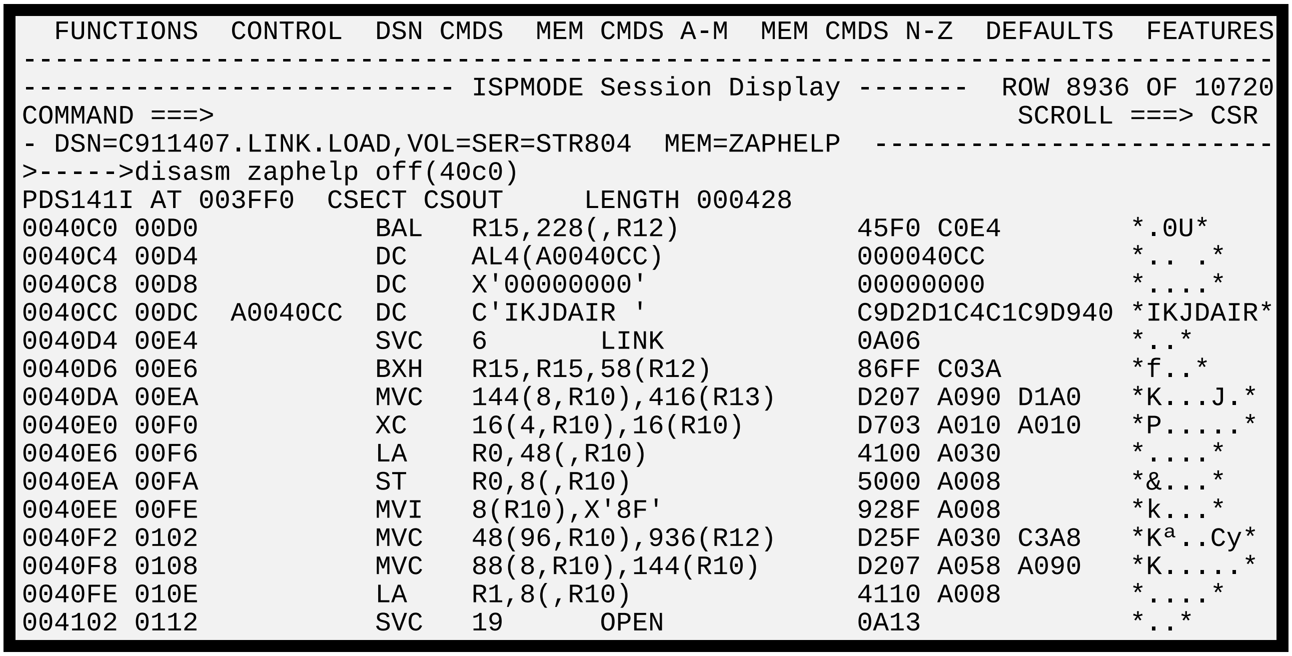
operation a one to five byte reconstructed operation code.

operands the reconstructed operand values.

comments for SVC calls, the original MACRO (for example, GETMAIN or TPUT).

hex representation one to three columns of the original hexadecimal data.

character representation one to 8 bytes of the same data in character mode surrounded by asterisks.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display -------  ROW 8936 OF 10720
COMMAND ===>                                                   SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAPHELP  ------------------------
>----->disasm zaphelp off(40c0)
PDS141I AT 003FF0  CSECT CSOUT     LENGTH 000428
0040C0 00D0           BAL   R15,228(,R12)          45F0 C0E4        *.0U*
0040C4 00D4           DC    AL4(A0040CC)           000040CC         *.. .*
0040C8 00D8           DC    X'00000000'            00000000         *....*
0040CC 00DC  A0040CC  DC    C'IKJDAIR '            C9D2D1C4C1C9D940 *IKJDAIR*
0040D4 00E4           SVC   6       LINK           0A06             *..*
0040D6 00E6           BXH   R15,R15,58(R12)        86FF C03A        *f..*
0040DA 00EA           MVC   144(8,R10),416(R13)    D207 A090 D1A0   *K...J.*
0040E0 00F0           XC    16(4,R10),16(R10)      D703 A010 A010   *P.....*
0040E6 00F6           LA    R0,48(,R10)            4100 A030        *....*
0040EA 00FA           ST    R0,8(,R10)             5000 A008        *&...*
0040EE 00FE           MVI   8(R10),X'8F'           928F A008        *k...*
0040F2 0102           MVC   48(96,R10),936(R12)    D25F A030 C3A8   *Kª..Cy*
0040F8 0108           MVC   88(8,R10),144(R10)     D207 A058 A090   *K.....*
0040FE 010E           LA    R1,8(,R10)             4110 A008        *....*
004102 0112           SVC   19      OPEN           0A13             *..*
```

Figure 35. Sample DISASM Subcommand

DISPLAY Subcommand

Purpose The DISPLAY subcommand lists member names from the data set directory. Note: the default member group is not affected by either the DISPLAY or PATTERN subcommands.

Example DISPLAY aa b4

Syntax

```
DISPLAY   name1 name2
```

Aliases D, DI, DIS, DISP, DISPL, DISPLA, DISPLAY

Defaults entire directory

Required none

Operands

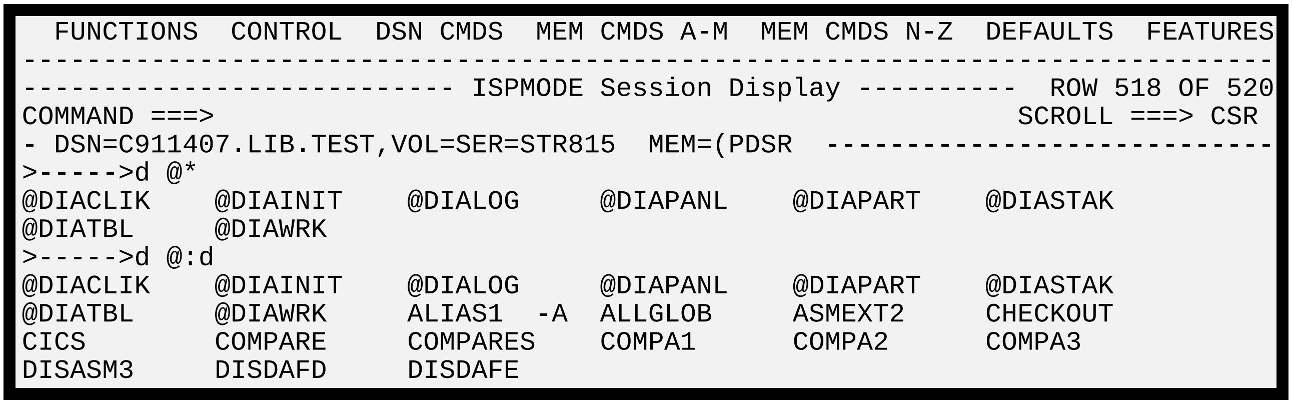
name1 specifies the starting member name (or portion of the member name) at which the display is to start. If omitted, the entire directory is displayed.

Note: a group name descriptor such as **part1:part2**, **seg1/seg2** or **part1*seg1** may be entered for name1, in which case, the output will consist of the member names which would be displayed by the MEMBERS subcommand.

name2 specifies the ending member name (or portion of the member name) after which the display is to stop. If omitted, the display continues to the end of the directory.

Remarks Neither DISPLAY nor PATTERN affect the current member group. If DISPLAY is entered without operands, all members in the data set are displayed. This differs from PATTERN in that PATTERN remembers its previously entered operand(s).

Note that **-A** is added following a displayed member name if the member name is an alias. The member name is listed in a combined hexadecimal/character format if it does not contain upper case alphameric characters (including @, $ and #) or if the first character is numeric. Unprintable characters (for 3270-type devices) in a member name are displayed as periods.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 518 OF 520
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=(PDSR  ----------------------------
>----->d @*
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK
>----->d @:d
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK     ALIAS1  -A  ALLGLOB     ASMEXT2     CHECKOUT
CICS        COMPARE     COMPARES    COMPA1      COMPA2      COMPA3
DISASM3     DISDAFD     DISDAFE
```

Figure 36. Sample DISPLAY Subcommand

DSAT Subcommand

This interface is optional, it should only be used if your installation has installed the public domain DSAT command. For current information on the DSAT command, refer to the CBT web site at http://www.cbttape.org file 112.

Purpose The DSAT subcommand is used to display allocation information for data sets on a direct access device.

DSAT will search the catalog for the entries for the data sets specified. Allocation information will be obtained from the volume table of contents, formatted and displayed. If a name is an index name, all data sets below the index will be displayed.

You may bypass the catalog search by supplying the volume name on which the data set resides. This option permits displaying information for uncataloged data sets.

Example DSAT lib

Syntax

```
DSAT [(dsnamelist)]   [ ALL/DAONLY                           ]
                      [ ALLOC/NOALLOC                        ]
                      [CCHHR/ NOCCHHR                        ]
                      [ CRDATE/NOCRDATE                      ]
                      [ DCB/NODCB                            ]
                      [DEVICE/ NODEVICE                      ]
                      [DEVTYPE/ NODEVTYPE                    ]
                      [ DSORG/NODSORG                        ]
                      [EXDATE/ NOEXDATE                      ]
                      [GDGDATA/ NOGDGDATA                    ]
                      [ GENERIC/DSONLY                       ]
                      [HARDCOPY(ddname)                      ]
                      [ HEADER/NOHEADER                      ]
                      [LASTREF /NOLASTREF                    ]
                      [PDS/ NOPDS                            ]
                      [ PRINT/NOPRINT                        ]
                      [RC(TALLOC/TUSED/TDIFF/                ]
                          LALLOC/LUSED/LDIFF/NUM/            ]
                          PREVIOUS/DSORG/DIRALLOC/           ]
                          DIRUSED/ENTRIES/ALIASES/MEMBERS)   ]
                      [SECONDARY/ NOSECONDARY                ]
                      [SEQNO/ NOSEQNO                        ]
                      [ SERIAL/NOSERIAL                      ]
                      [ TOTALS/NOTOTALS                      ]
                      [VOLUME(name)                          ]
```

Aliases DSA, DSAT

Defaults dsnamelist defaults to 'prefix' to list a user's data sets; see underscores above.
Note: if HARDCOPY is specified, NOPRINT is the default

Required none

Operands

dsnamelist one or more data set or index level names. TSO naming conventions are used. If a specified name is an index name, data sets under that level will be displayed. If no name is specified, the prefix will be used. If no name is specified and the profile specifies NOPREFIX, the userid will be used.

Note: if a keyword is entered, dsnamelist must be specified. If dsnamelist is omitted, it will not default to the prefix, but the keyword will be interpreted as the dsnamelist, not a keyword.

ALL display all data set names under an index level.
DAONLY display data sets on mounted direct access devices.
ALLOC list allocation information (tracks allocated, tracks free, and extents).
NOALLOC do not list allocation information.
CCHHR display Format 1 DSCB address.
NOCCHHR do not display CCHHR.
CRDATE list creation date.
NOCRDATE do not list creation date.
DCB list DCB information (RECFM, BLKSIZE, and LRECL).
NODCB do not list DCB information.
DEVTYPE display device type code (for example, 3010200F for a 3390).
NODEVTYPE do not display device type code.
DEVICE list device name (3380, 9345, etc.).
NODEVICE do not list device name.
DSORG list data set organization.
NODSORG do not list data set organization.
EXDATE list expiration date.
NOEXDATE do not list expiration date.
GDGDATA display information on generation data group indexes.
NOGDGDATA do not display information on generation data group indexes.
GENERIC name specifies a generic key.
DSONLY treat names as data set names, not index names.
HARDCOPY(dd) specifies the DDNAME of a pre-allocated data set for output from DSAT.
HEADER display a output header line.
NOHEADER do not display a header line.
LASTREF display last date referenced
NOLASTREF do not display last date referenced.
PDS list PDS directory information (blocks alloc, used, entries, and aliases).
NOPDS do not list PDS directory information.
PRINT output is to be displayed.
NOPRINT output is not to be displayed, except for error messages.
RC(type) specifies how the return code is to be set.

Note that if DSAT is used as a STARWARP subcommand, return code processing is not performed (the return code will be set to zero). However, if you use DSAT in a CLIST or rexx exec or in single subcommand mode as in the following example, the return code will be set properly:

```
STARWARP FILE(ISPPROF) DSAT LIB RC(TALLOC)
```

If RC is not specified, RC=4 means successful completion and RC=16 means unable to interpret command Otherwise, RC is set to a value as specified by any of the following:

TALLOC total tracks allocated
TUSED total tracks used
TDIFF total overallocation (TALLOC minus TUSED)
LALLOC tracks allocated by last data set

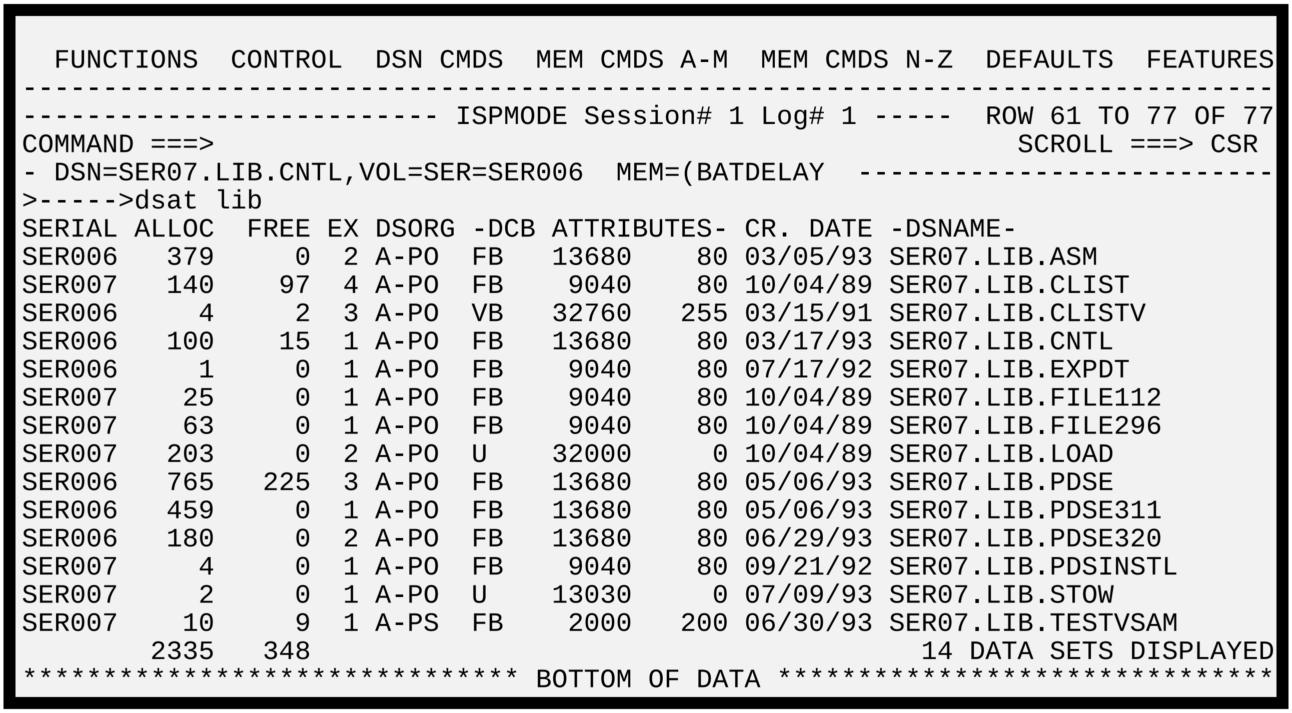
LUSED tracks used by last data set
LDIFF last data set overallocation (LALLOC minus LUSED)
NUM number of data sets displayed (includings data sets not found)
PREVIOUS assumes the return code of the previous command.
DSORG sets the return code based on the data set organization of the data sets displayed as follows:
1 All data sets were sequential
2 All data sets were PDS
3 Data sets were mixed PDS and sequential
4 Data set other than PDS or sequential or an error.
DIRALLOC number of directory blocks allocated.
DIRUSED number of directory blocks used.
ENTRIES number of entries in the directory (members and aliases).
ALIASES the number of alias entries.
MEMBERS number of entries less the number of aliases.

SECONDARY display secondary allocation information.
NOSECONDARY do not display secondary allocation information.
SEQNO list file sequence numbers.
NOSEQNO do not list file sequence numbers.
SERIAL list volume names.
NOSERIAL do not list volume names.
TOTALS display totals (tracks allocated, tracks free, and data sets displayed).
NOTOTALS do not display totals.
VOLUME specifies the volume to be searched for the data set.

Remarks DSAT may be used in CLISTs to find the allocation of a data set or a group of data sets and set the return code to the specified value. The return code may then be tested with the WHEN command. Output may be suppressed by specifying NOPRINT. The HARDCOPY keyword may be used to direct the output to a pre-allocated data set.

The attributes to be displayed may be selected by specifying keyword operands. The information that may be displayed is:

1. Volume name on which the data set is located.
2. File sequence number.
3. Device type code from catalog entry.
4. Allocation (allocated, free, and extents).
5. Secondary allocation (amount and units).
6. Data set organization.
7. DCB (RECFM, BLKSIZE, and LRECL).
8. Creation date.
9. Expiration date.
10. Last date referenced
11. Fully qualified data set name.
12. CCHHR of the Format 1 DSCB.
13. Generation data group data.
14. PDS directory information.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1 -----  ROW 61 TO 77 OF 77
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  --------------------------
>----->dsat lib
SERIAL ALLOC  FREE EX DSORG -DCB ATTRIBUTES- CR. DATE -DSNAME-
SER006   379      0  2 A-PO  FB   13680    80 03/05/93 SER07.LIB.ASM
SER007   140     97  4 A-PO  FB    9040    80 10/04/89 SER07.LIB.CLIST
SER006     4      2  3 A-PO  VB   32760   255 03/15/91 SER07.LIB.CLISTV
SER006   100     15  1 A-PO  FB   13680    80 03/17/93 SER07.LIB.CNTL
SER006     1      0  1 A-PO  FB    9040    80 07/17/92 SER07.LIB.EXPDT
SER007    25      0  1 A-PO  FB    9040    80 10/04/89 SER07.LIB.FILE112
SER007    63      0  1 A-PO  FB    9040    80 10/04/89 SER07.LIB.FILE296
SER007   203      0  2 A-PO  U    32000     0 10/04/89 SER07.LIB.LOAD
SER006   765    225  3 A-PO  FB   13680    80 05/06/93 SER07.LIB.PDSE
SER006   459      0  1 A-PO  FB   13680    80 05/06/93 SER07.LIB.PDSE311
SER006   180      0  2 A-PO  FB   13680    80 06/29/93 SER07.LIB.PDSE320
SER007     4      0  1 A-PO  FB    9040    80 09/21/92 SER07.LIB.PDSINSTL
SER007     2      0  1 A-PO  U    13030     0 07/09/93 SER07.LIB.STOW
SER007    10      9  1 A-PS  FB    2000   200 06/30/93 SER07.LIB.TESTVSAM
        2335    348                                      14 DATA SETS DISPLAYED
******************************* BOTTOM OF DATA *******************************
```

Figure 37. Sample DSAT Subcommand

DSNAME Subcommand

Purpose The DSNAME subcommand displays statistics on the current data set allocation using message format PDS200I, PDS210I or PDS220I depending on the last DSNAME operand (MSG, TSO, or JCL) entered.

Notes: the DSNAME subcommand is automatically invoked while executing several subcommands such as in CHANGE and USAGE. MSG format output is always used to display VSAM data sets.

Example DSN

Syntax

```
DSNAME   [MSG / TSO / JCL]
```

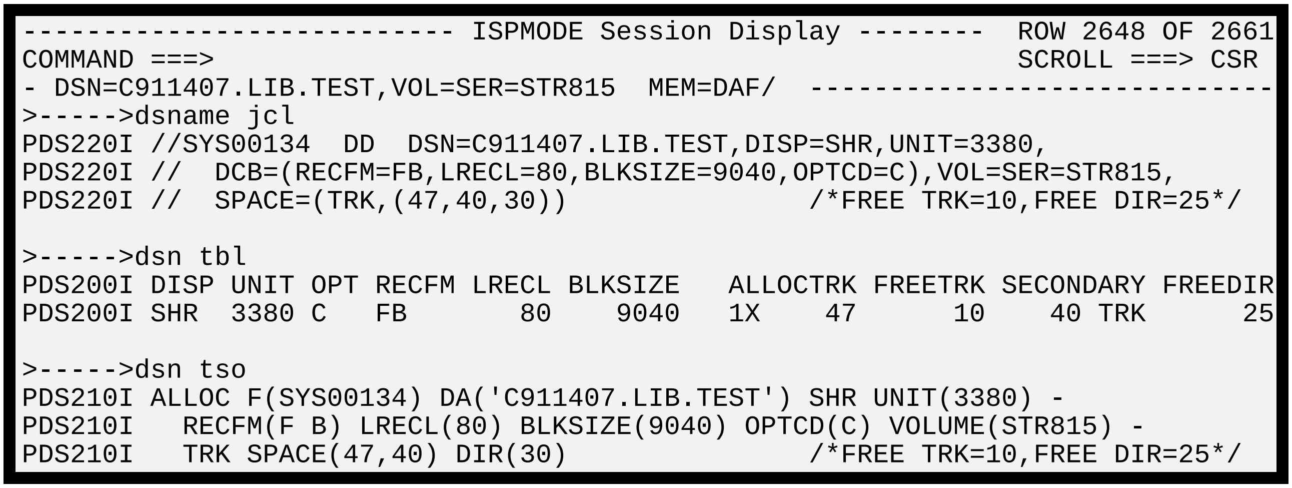
Aliases DS, DSN, DSNA, DSNAM, DSNAME, DSNAMES

Defaults initially MSG; thereafter, previous MSG/TSO/JCL

Required none

Operands

MSG use the PDS200I message (table or msg format) for allocation display.
TABLE use the PDS200I message (table or msg format) for allocation display.
TSO use the PDS210I message (TSO format) for allocation display.
JCL use the PDS220I message (JCL or batch format) for allocation display.
BATCH use the PDS220I message (JCL or batch format) for allocation display.



```
--------------------------- ISPMODE Session Display --------  ROW 2648 OF 2661
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=DAF/  -----------------------------
>----->dsname jcl
PDS220I //SYS00134  DD  DSN=C911407.LIB.TEST,DISP=SHR,UNIT=3380,
PDS220I //  DCB=(RECFM=FB,LRECL=80,BLKSIZE=9040,OPTCD=C),VOL=SER=STR815,
PDS220I //  SPACE=(TRK,(47,40,30))               /*FREE TRK=10,FREE DIR=25*/

>----->dsn tbl
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80    9040   1X    47      10    40 TRK      25

>----->dsn tso
PDS210I ALLOC F(SYS00134) DA('C911407.LIB.TEST') SHR UNIT(3380) -
PDS210I   RECFM(F B) LRECL(80) BLKSIZE(9040) OPTCD(C) VOLUME(STR815) -
PDS210I   TRK SPACE(47,40) DIR(30)               /*FREE TRK=10,FREE DIR=25*/
```

Figure 38. Sample DSNAME Subcommand

DVOL Subcommand

This interface is optional, it should only be used if your installation has installed the public domain DVOL command. For current information on the DVOL command, refer to the CBT web site at **http://www.cbttape.org**.

Purpose The DVOL subcommand is used to display the amount of of free space on a direct access device.

Example DVOL tso long

Syntax

```
DVOL [(volumelist)/* ] [SHORT/LONG        ]
                        [HEADER/NOHEADER ]
                        [PRINT/NOPRINT    ]
                        [SERONLY          ]
                        [HARDCOPY(ddname) ]
```

Aliases DV, DVO, DVOL

Defaults * to display all volumes, SHORT, HEADER and PRINT.
Note: if HARDCOPY is specified, NOPRINT is the default.

Required none

Operands

volumelist is a list of one or more volume names of mounted direct access volumes or partial volume names.

If no volume is entered, or if an * is specified, the information is displayed for each direct access volume on the system.

If a volume name is entered with a U followed by one to three digits, the data is also used as a UCB name.

SHORT display short form output.
LONG display long form output.
HEADER display a short form header.
NOHEADER display no short form header.
PRINT display output.
NOPRINT display no output.
SERONLY display only a list of mounted volumes.
HARDCOPY(dd) specifies the DDNAME of a pre-allocated data set to receive a copy of DVOL output.

DVOL Subcommand

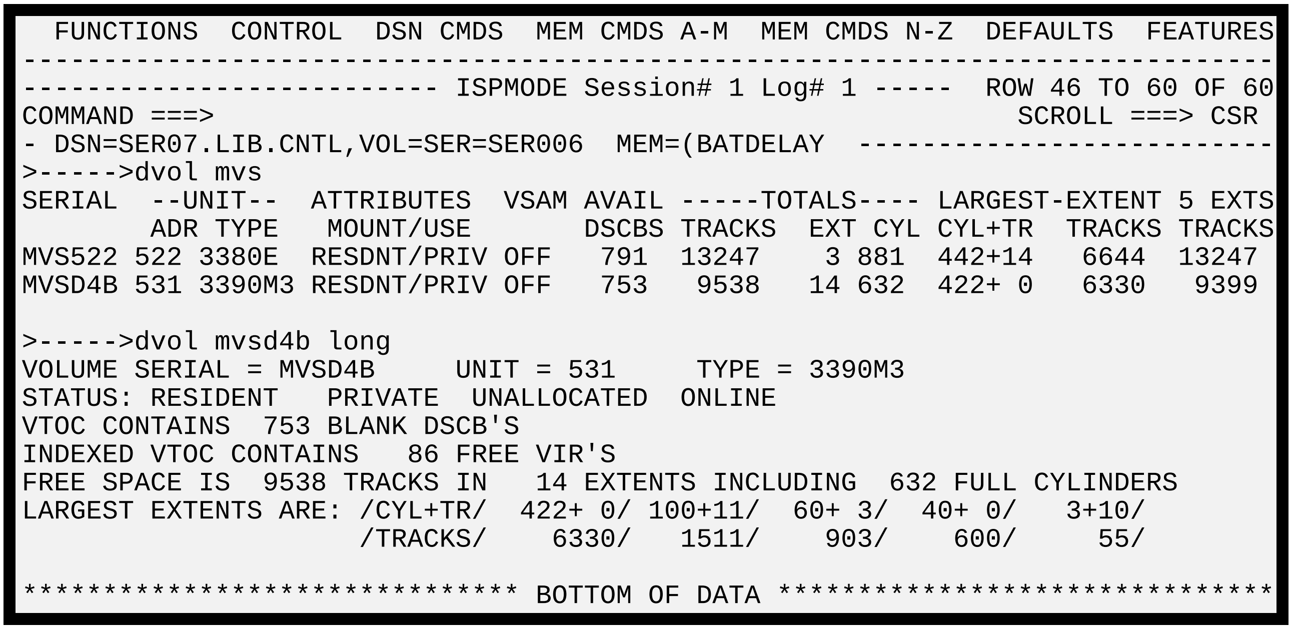
Remarks DVOL will read the Format 4 and Format 5 DSCB's from the VTOC of a direct access volume and display:

1. Volume name
2. Unit address
3. Device type including density indicators
4. Mount status and use status
5. Number of blank DSCB's in the VTOC
6. Condition of the VTOC indicators byte
7. VSAM data fields
8. Total free space in tracks
9. Number of free extents
10. Number of free cylinders
11. Size of largest extents (up to 5) in cylinders + tracks
12. Size of largest extents (up to 5) in tracks

The Return code is set to the total number of tracks in the 5 largest extents (to a maximum of 4095) for the last volume displayed.

If an error condition exists on the volume, the return code will be set to 0.

This interface is optional, it should only be used if your installation has installed the public domain DVOL command.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1 -----  ROW 46 TO 60 OF 60
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  --------------------------
>----->dvol mvs
SERIAL  --UNIT--  ATTRIBUTES  VSAM AVAIL -----TOTALS---- LARGEST-EXTENT 5 EXTS
        ADR TYPE   MOUNT/USE       DSCBS TRACKS  EXT CYL CYL+TR  TRACKS TRACKS
MVS522 522 3380E  RESDNT/PRIV OFF   791  13247    3 881  442+14   6644  13247
MVSD4B 531 3390M3 RESDNT/PRIV OFF   753   9538   14 632  422+ 0   6330   9399

>----->dvol mvsd4b long
VOLUME SERIAL = MVSD4B     UNIT = 531     TYPE = 3390M3
STATUS: RESIDENT   PRIVATE  UNALLOCATED  ONLINE
VTOC CONTAINS  753 BLANK DSCB'S
INDEXED VTOC CONTAINS   86 FREE VIR'S
FREE SPACE IS  9538 TRACKS IN   14 EXTENTS INCLUDING  632 FULL CYLINDERS
LARGEST EXTENTS ARE: /CYL+TR/  422+ 0/ 100+11/  60+ 3/  40+ 0/   3+10/
                     /TRACKS/    6330/   1511/    903/    600/     55/

******************************* BOTTOM OF DATA *******************************
```

Figure 41. Sample DVOL Subcommand

EDIT Subcommand

Purpose The EDIT or SPFEDIT subcommand enters ISPF edit for a member.

Example EDIT mema:memb

Syntax

```
EDIT   memgroup [OLD/NEW                       ]
                [MACRO(macname)                ]
                [SM(macname)/SETMACRO(macname) ]
                [PROFILE(profname)             ]
                [SP(profname)/SETPROF(profname)]
                [CONFIRM/NOCONFIRM             ]
                [SETCONFIRM/NOSETCONFIRM       ]
              * [FROMKEY(key)/FROMADDRESS(add)/FROMNUMBER(num)  ]
              * [TOKEY(key)/TOADDRESS(add)/TONUMBER(num)        ]
*NOTE: Lines with an asterisk are supported for VSAM with PEDIT or EDIF.
```

Aliases E, ED, EDI, EDIT, S, SP, SPF, SPFE, SPFED, SPFEDI, SPFEDIT

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults memgroup, confirm

Required none

Operands

memgroup identifies the source member(s) to be edited.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

OLD verify that the member exists before invoking ISPF EDIT service.

NEW verify that the member does not exist before invoking ISPF EDIT service.

MACRO(mac) specifies an initial edit macro name.

SM(mac) specifies a default initial edit macro name; to nullify this parameter, you may enter **SM(*)**.

SETMACRO(mac) specifies a default initial edit macro name; to nullify this parameter, you may enter **SETMACRO(*)**.

PROFILE(prof) specifies the edit profile name (the supplied name will be used instead of the low-level DSNAME qualifier).

SP(prof) specifies a default edit profile name; to nullify this parameter, you may enter **SP(*)**.

SETPROF(prof) specifies a default edit profile name; to nullify this parameter, you may enter **SETPROF(*)**.

CONFIRM for ISPF 4.x, specifies that you desire prompting for MOVE, REPLACE or CANCEL operations.

NOCONFIRM for ISPF 4.x., specifies that you do not want confirmation prompting for MOVE, REPLACE or CANCEL operations.

SETCONFIRM for ISPF 4.x, specifies a default of **CONFIRM** for subsequent edits in this STARWARP session. You can override this keyword on individual edit commands with **NOCONFIRM**.

NOSETCONFIRM for ISPF 4.x, specifies a default of **NOCONFIRM** for subsequent edits in this STARWARP session. You can override this keyword on individual edit sessions with **CONFIRM**.

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access begins at the first record whose key matches (or is greater than) the portion of the key specified.

This parameter may be used with TOKEY and it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.

This parameter may be used with TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified.

This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.

This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

Remarks On the first entry of an EDIT or SPFEDIT subcommand, STARWARP will present any pending EDIT recovery sessions to you before entering EDIT for a selected member.

If **:** is entered for the member name position, a MEMLIST will be provided instead.

VSAM data sets are normally edited with the PEDIT command. However, you may use ISPF EDIF services to access up to 255 characters of any individual record but note that the SAVE command is disabled. A different interface may have been choosen during STARWARP installation; of the interfaces supported, only PEDIT and EDIF support positioning keywords such as FROMKEY. To determine which interface is being used, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "EDIT calls" with one of the following values:

EDIF	this uses the ISPF EDIF interface with up to 255 characters per record
PEDIT	this uses PEDIT services.
%VSAMMED	this uses MacKinney System's VSAM Utility Edit

EDREC Subcommand

Purpose The EDREC subcommand explicitly invokes ISPF edit recovery.

Example EDREC

Syntax

```
EDREC
```

Aliases EDR, EDRE, EDREC

Operands (no operands are supported for the EDREC subcommand).

Remarks This subcommand is provided to allow you to control when EDIT RECOVERY is to take place. Normally, EDIT RECOVERY procedures would be initiated on the first EDIT subcommand; however, with the EDREC subcommand you can choose to perform the appropriate action before the first EDIT subcommand.

ENCODE Subcommand

Purpose The ENCODE subcommand encrypts a member. This was written using the data encryption algorithm submitted by IBM to the National Bureau of Standards and published in the Federal Register.

Example ENCODE   original mymember keyexamp

Syntax

```
ENCODE input output key [SHR/OLD       ]
```

Aliases ENC, ENCO, ENCOD, ENCODE

Defaults OLD

Required input, output, key

Operands

input identifies the member to be encoded.

If this is a 1-8 byte simple name it is taken to be a member name in the current data set; otherwise it is taken to be data.set(member).

output identifies the receiving member.

If this is a 1-8 byte simple name it is taken to be a member name in the current data set; otherwise it is taken to be data.set(member).

key This is a 1-8 byte alphameric key which is to be used for encoding the data set member.

OLD This allocation is not to be shared with any other allocation for this data set.

SHR This allocation is to be shared with other allocations for this data set.

Remarks This subcommand encrypts data using the supplied translation key. Since this algorithm works on eight bytes at a time, the actual number of bytes enciphered or deciphered will be an integral number of eight bytes that is less than or equal to the record length. Note that the greatest number of bytes left unchanged on a record would be the rightmost seven bytes.

Warning: this method does not record your key. If you cannot provide the key, the data will not be readable. Note that the ENCODE load module may be used independently of STARWARP. A HELP entry is available but its basic syntax is as follows:

ENCODE input.dsn(memin) output.dsn(memout) key SHR/OLD

END Subcommand

Purpose The END subcommand terminates the STARWARP command. Note: in ISPMODE, the END subcommand terminates the current function; multiple END subcommands are usually required to terminate the STARWARP command.

Example END

Syntax

```
END
```

Aliases EN, END

Operands (no operands are supported for the END subcommand).

Remarks In ISPMODE or MEMLIST an END will usually terminate the current function instead of the STARWARP command. If you want to terminate STARWARP you may enter repeated END commands or a QUIT subcommand.

EXCLUDE Subcommand

Purpose The EXCLUDE subcommand removes member names from the current MEMLIST. EXCLUDE may be used by itself or as follows:

```
 FIND isp* 'string' else(exclude)
     - or -
 IF     mem/ alias .... then(exclude)
```

Example EXCLUDE   isp* norent

Syntax

```
EXCLUDE memgroup
        [SINCE/BEFORE                         ]
        [TODAY/YESTERDAY/WEEK/CURRENT/BIWEEK/
           MONTH/QUARTER/HALFYEAR/YEAR/BIYEAR/
           LAST(numdays)/DATE(yyyy/mm/dd)     ]
        [CHANGED(yyyy/mm/dd:yyyy/mm/dd)       ]
        [CREATED(yyyy/mm/dd:yyyy/mm/dd)       ]
        [ABOVE(Count1)                        ]
        [ALIAS/NOALIAS                        ]
        [AMODE24/AMODE31/AMODEANY/NOAMODE24/
           NOAMODE31/NOAMODEANY               ] (load only)
        [AUTH/NOAUTH                          ] (load only)
        [BELOW(Count2)                        ]
        [DC/NODC                              ] (load only)
        [EDIT/NOEDIT                          ] (load only)
        [EXEC/NOEXEC                          ] (load only)
        [FLEVEL/NOFLEVEL                      ] (load only)
        [HASALIAS/NOHASALIAS                  ]
        [ID(Puid)/NOID/NOTID(Puid)            ]
        [LKED(Partl)                          ] (load only)
        [LOADONLY/NOLOADONLY                  ] (load only)
        [MODULE({* / Fullm / Partm*)}         ] (load only)
        [NULL/NONULL                          ]
        [ORPHAN/NOORPHAN                      ]
        [OVERLAY/NOOVERLAY                    ] (load only)
        [PAGE/NOPAGE                          ] (load only)
        [REFR/NOREFR                          ] (load only)
        [RENT/NORENT                          ] (load only)
        [REUS/NOREUS                          ] (load only)
        [RMODE24/RMODEANY/NORMODE24/NORMODEANY] (load only)
        [SCTR/NOSCTR                          ] (load only)
        [SSI(hxdata)/SSI/NOSSI/PARTSSI(hxdata)]
        [SYSMOD(Partu) / USERDATA(Partu)      ] (load only)
        [TEST/NOTEST                          ] (load only)
        [TRANS(Partt)                         ] (load only)
        [TTR(Lttr:Httr)                       ]
        [USERDATA(Partu) / SYSMOD(Partu)      ] (load only)
        [USERID(Puid)/NOUSERID/NOTUSERID(Puid)]
        [VSLKED/NOVSLKED                      ] (load only)
        [ZAP(Partz)                           ] (load only)
```

Aliases EXC, EXCL, EXCLU, EXCLUD, EXCLUDE

Defaults memgroup, SINCE

EXCLUDE Subcommand

Required none

Operands

memgroup identifies the member(s) to be excluded from the current MEMLIST.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

SINCE use dates from the specified date to the current date. SINCE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

FROM use dates from the specified date to the current date. FROM indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

BEFORE use dates before (and including) the specified date. BEFORE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

TO use dates before (and including) the specified date. TO indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

TODAY a date, equivalent to LAST(0)

YESTERDAY a date, equivalent to LAST(1)

WEEK a date, equivalent to LAST(7)

CURRENT a date, equivalent to LAST(10)

BIWEEK a date, equivalent to LAST(14)

MONTH a date, equivalent to LAST(30)

QUARTER a date, equivalent to LAST(120)

HALFYEAR a date, equivalent to LAST(183)

YEAR a date, equivalent to LAST(365)

BIYEAR a date, equivalent to LAST(730)

LAST(numdays) a date, indicates the number of days before today.

DATE(cdate) a date, indicates the actual date; it may be entered in ISPF format (yyyy/mm/dd), or Julian format (yyyy.ddd).

CHANGED(date: excludes members modified by an ISPF editor or a ZAP program between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like:
CHANGED(yyyy/mm/dd:yyyy/mm/dd)

Note that only members with ISPF statistics or members created by a linkage editor are supported.

CREATED(date: excludes members created with ISPF statistics or by the linkage editor between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like:
CREATED(yyyy/mm/dd:yyyy/mm/dd)

ABOVE(Count1) exclude members with more than the number of lines (amount of storage for load modules) specified by the number, **Count1**.

ALIAS exclude alias members.

NOALIAS exclude main members.

AMODE24 exclude modules with addressing mode 24.

NOAMODE24 exclude modules with addressing mode 31 or ANY.

AMODE31 exclude modules with addressing mode 31.

NOAMODE31 exclude modules with addressing mode 24 or ANY.

AMODEANY exclude modules with addressing mode ANY.

NOAMODEANY exclude modules with addressing mode 24 or 31.

AUTH exclude APF authorized modules.

NOAUTH exclude non-authorized modules.

BELOW(Count2) exclude members with less than the number of lines (amount of storage for load modules) specified by the number, **Count2**.

DC exclude downward-compatible modules.

NODC exclude non-downward-compatible modules.

EDIT exclude modules which can be reprocessed by the linkage editor.

NOEDIT exclude modules which can not be reprocessed by the linkage editor.

EXEC exclude EXECUTABLE modules.

NOEXEC exclude non-EXECUTABLE modules.

FLEVEL exclude modules processed by the F-level linkage editor.

NOFLEVEL exclude modules not processed by the F-level linkage editor.

HASALIAS exclude main members which have one or more aliases.

NOHASALIAS exclude alias members or main members without any aliases.

ID(Puid) exclude members with ISPF statistics and userids matching the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. ID is an alias of USERID.

NOID exclude members without ISPF statistics. NOID is an alias of NOUSERID.

NOTID(Puid) exclude members without ISPF statistics or members with ISPF statistics and userids which do not match the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. NOTID is an alias of NOTUSERID.

LKED(Partl) exclude modules identified as linked by the linkage editor defined by the partial name, **Partl**. **Partl** may be entered as a partial linkage editor IDR name with one to ten characters.

LOADONLY exclude modules marked for LOAD ONLY.

NOLOADONLY exclude modules not marked for LOAD ONLY.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT names for attribute searches. Note that if this parameter is entered with any TRANS, SYSMOD/USERDATA or ZAP keywords, only those CSECTs which satisfy the MODULE keyword are checked for the other keyword conditions.

The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.

MODULE(Fullm) exclude any module containing a CSECT or ENTRY called FULLM.

MODULE(Partm*) exclude any module containing a CSECT or ENTRY called PARTM...

ORPHAN exclude alias members which have no associated main member.

NOORPHAN exclude main members or alias members which have an associated main member.

OVERLAY exclude OVERLAY modules.

NOOVERLAY exclude non-OVERLAY modules.

PAGE exclude modules marked for loading on a page boundary.

NOPAGE exclude modules not marked for loading on a page boundary.

REFR exclude refreshable modules.

NOREFR exclude non-refreshable modules.

RENT exclude reentrant modules.

NORENT exclude non-reentrant modules.

REUS exclude reusable modules.

NOREUS exclude non-reusable modules.

RMODE24 exclude modules with residence mode 24.

NORMODE24 exclude modules with residence mode ANY.

RMODEANY exclude modules with residence mode ANY.

NORMODEANY exclude modules with residence mode 24.

SCTR exclude scatter-loaded modules.

NOSCTR exclude non-scatter-loaded modules.

SSI(hxdata) exclude members with matching SSI data. Note that this is implemented as a generic search matching SSI characters from left to right for the number of characters entered.

SSI exclude members with SSI data.

NOSSI exclude members without SSI data.

PARTSSI(hxdata) exclude members with matching SSI data. Note that this is implemented as a pattern search matching SSI digits as a string anywhere in the SSI field of he member.

SYSMOD(Partu) exclude modules with user IDR data which matches the partial name, **Partu**. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the SYSMOD operand applies only to CSECT's whose names satisfy the MODULE argument.

TEST exclude modules with the TEST attribute.

NOTEST exclude modules without the TEST attribute.

TRANS(Partt) exclude modules with CSECTS identified as having been assembled or compiled by the translator defined by the partial name, **Partt**. **Partt** may be entered as a partial translator IDR name with one to ten characters. Note that if MODULE(...) is also entered, the TRANS operand applies only to CSECT's whose names satisfy the MODULE argument.

TTR(Lttr:Httr) exclude members whose start address is in the specified TTR range. **Lttr** defaults to 0 and may be entered as a hexadecimal TTR value from 0 through FFFFFF. **Httr** defaults to FFFFFF and may be entered as a hexadecimal TTR value from 0 through FFFFFF.

USERDATA(Partu) exclude modules with user IDR data which matches the partial name, **Partu**. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the USERDATA operand applies only to CSECT's whose names satisfy MODULE.

USERID(Puid) exclude members with ISPF statistics and userids matching the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. USERID is an alias of ID.

NOUSERID exclude members without ISPF statistics. NOUSERID is an alias of NOID.

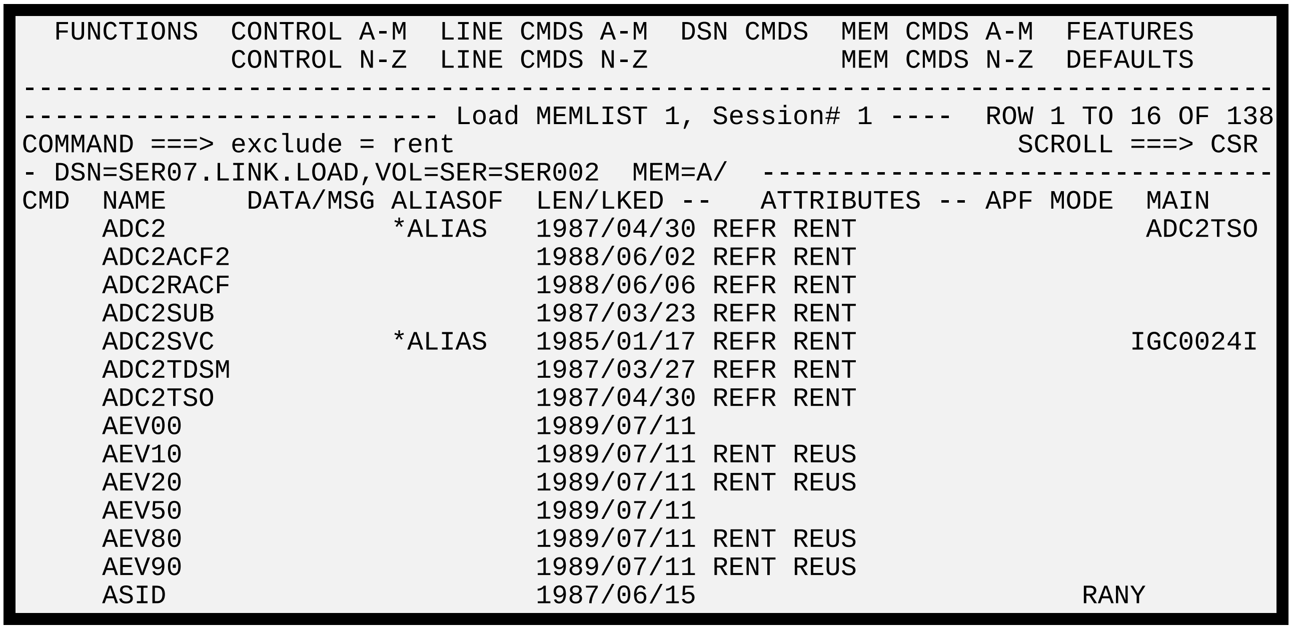
NOTUSERID(Puid) exclude members without ISPF statistics or members with ISPF statistics and userids which do not match the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. NOTUSERID is an alias of NOTID.

VSLKED exclude modules linked by the MVS OS/VS linkage editor.

NOVSLKED exclude modules not linked by the MVS OS/VS linkage editor.

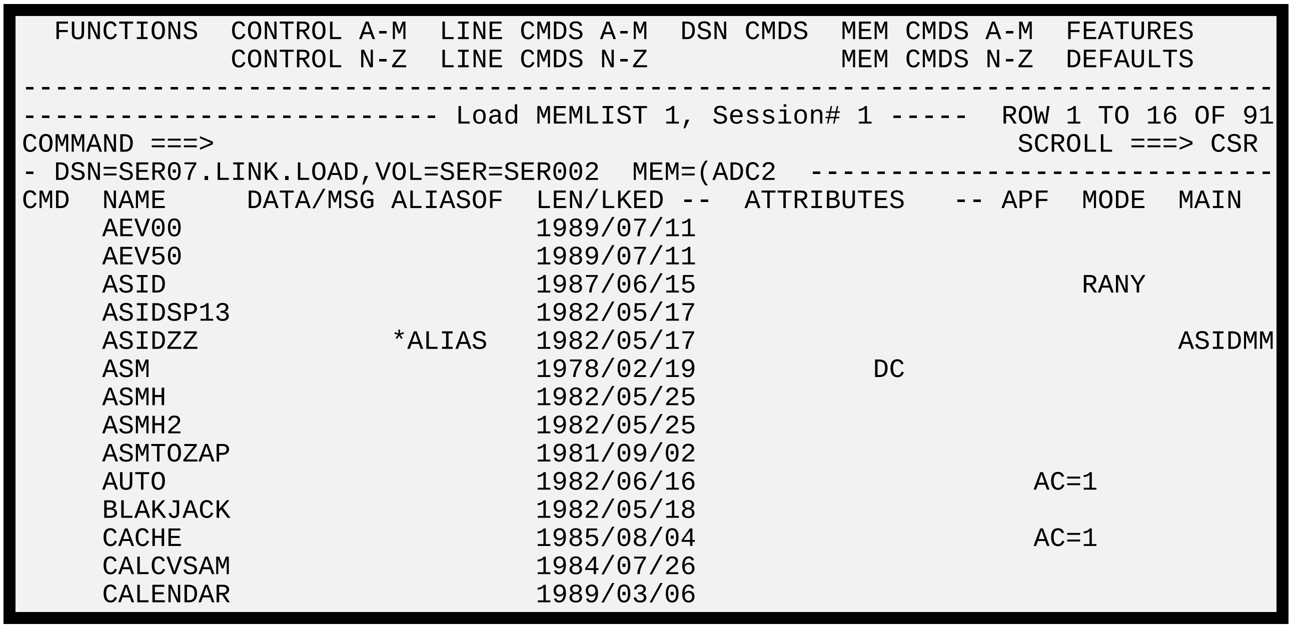
ZAP(Partz) exclude modules with zap IDR data which matches the partial name, **Partz**. **Partz** may be entered as partial zap IDR data with one to eight characters. Note that if MODULE(...) is also entered, the ZAP operand applies only to CSECT's whose names satisfy the MODULE argument.

Remarks If a member to be excluded is not in the MEMLIST, no action is taken. Note: the following sample screens show a before image of a member list and an after image showing the results of a **EXCLUDE = RENT**



```
  FUNCTIONS  CONTROL A-M  LINE CMDS A-M  DSN CMDS  MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z            MEM CMDS N-Z  DEFAULTS
------------------------------------------------------------------------------
------------------------- Load MEMLIST 1, Session# 1 ----  ROW 1 TO 16 OF 138
COMMAND ===> exclude = rent                                   SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=A/  --------------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED --   ATTRIBUTES -- APF MODE  MAIN
     ADC2              *ALIAS   1987/04/30 REFR RENT                  ADC2TSO
     ADC2ACF2                   1988/06/02 REFR RENT
     ADC2RACF                   1988/06/06 REFR RENT
     ADC2SUB                    1987/03/23 REFR RENT
     ADC2SVC           *ALIAS   1985/01/17 REFR RENT                 IGC0024I
     ADC2TDSM                   1987/03/27 REFR RENT
     ADC2TSO                    1987/04/30 REFR RENT
     AEV00                      1989/07/11
     AEV10                      1989/07/11 RENT REUS
     AEV20                      1989/07/11 RENT REUS
     AEV50                      1989/07/11
     AEV80                      1989/07/11 RENT REUS
     AEV90                      1989/07/11 RENT REUS
     ASID                       1987/06/15                        RANY
```

Figure 42. Sample EXCLUDE Subcommand (Before)



```
  FUNCTIONS  CONTROL A-M  LINE CMDS A-M  DSN CMDS  MEM CMDS A-M  FEATURES
             CONTROL N-Z  LINE CMDS N-Z            MEM CMDS N-Z  DEFAULTS
------------------------------------------------------------------------------
------------------------- Load MEMLIST 1, Session# 1 -----  ROW 1 TO 16 OF 91
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=(ADC2  ----------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED --  ATTRIBUTES   -- APF  MODE  MAIN
     AEV00                      1989/07/11
     AEV50                      1989/07/11
     ASID                       1987/06/15                        RANY
     ASIDSP13                   1982/05/17
     ASIDZZ            *ALIAS   1982/05/17                              ASIDMM
     ASM                        1978/02/19          DC
     ASMH                       1982/05/25
     ASMH2                      1982/05/25
     ASMTOZAP                   1981/09/02
     AUTO                       1982/06/16                     AC=1
     BLAKJACK                   1982/05/18
     CACHE                      1985/08/04                     AC=1
     CALCVSAM                   1984/07/26
     CALENDAR                   1989/03/06
```

Figure 43. Sample EXCLUDE Subcommand (After)

EXEC Subcommand

Purpose The EXEC or % subcommand executes a CLIST containing STARWARP subcommands. The TSO CLIST processor is used; both implicit (%...) and explicit (EXEC ...) call forms are supported.

Example EXEC   lib(clistmem) 'parms' list

Syntax

```
{EXEC   clistname ['operands'] / %clistmem [operands]}
```

Aliases %, EX, EXE, EXEC

Defaults none

Required clistname / clistmem

Operands

clistname specifies the CLIST data set name.

clistmem specifies the name of a member in the CLIST partitioned data set allocated to FILE(SYSPROC).

operands optional, may include any CLIST parameters.

Remarks The % (implied CLIST) and EXEC subcommands are supported in an ISPF dialog environment; you may want to use the ISPF **TSO** command to invoke a CLIST which operates under ISPF (independent of STARWARP).

When subcommands are entered from a CLIST (or in batch mode or from a storage stack), YES/NO prompts are not provided; instead, YES responses are assumed in each case. Note: YES responses are also assumed if **CONTROL NOPROMPT** is in effect.

FILTER Function

Purpose The FILTER command can be used to set selection values for use later by the LISTC/LISTF or MASK function. The NOFILTER keyword in MASK will reset all FILTER options to their defaults.

Example FILTER novs

Syntax

```
FILTER  [ALL/P/PS/PO/PE/PDS/DA/IS/NOVS/VS/GDG]
        [ASSOC/NOASSOC                       ]
        [CHECKCAT/NOCHECK                    ]
        [MIGRAT/NOMIGRAT/ONLYMIGR            ]
        [OPTICAL/NOOPTICAL                   ]
        [QUICK/NOQUICK                       ]
        [RESET                               ]
```

Aliases FIL, FILT,\ FILTE, FILTER

Defaults ALL, ASSOC, NOCHECK, MIGRAT, NOOPTICAL, NOQUICK

Required none

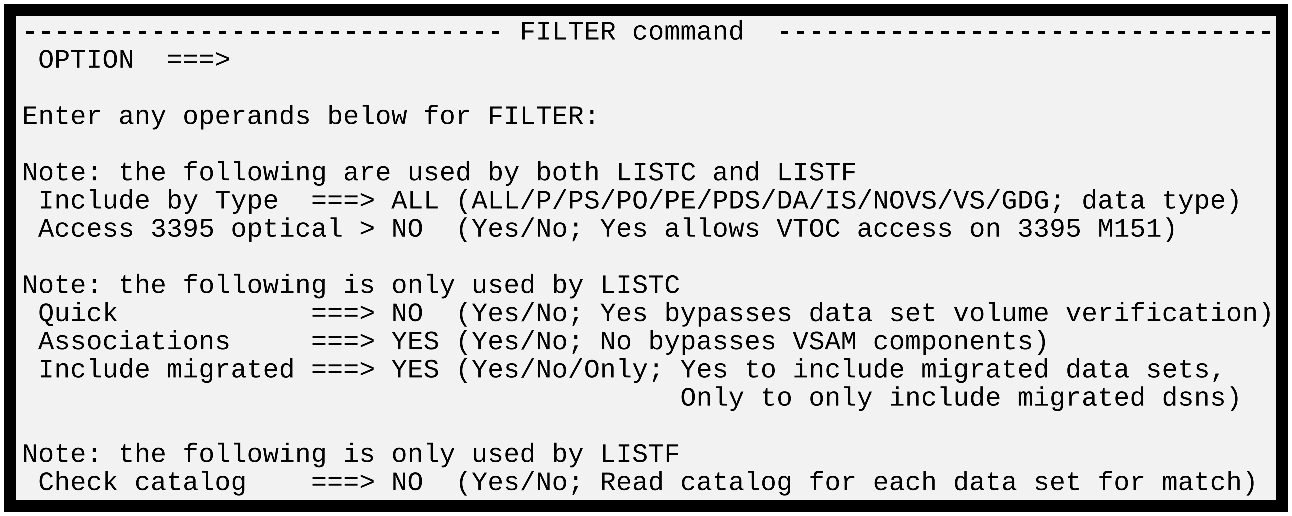
Operands

ALL	default, searches for all data set types
P	searches for PS, PO and PE data sets
PS	searches for sequential data sets
PO	searches for partitioned data sets
PE	searches for partitioned extended data sets
PDS	searches for PO and PE data sets
DA	searches for direct data sets
IS	searches for ISAM data sets
NOVS	searches for non-VSAM data sets
VS	searches for VSAM data sets
GDG	searches for Generation data sets
ASSOC	default for LISTC, adds associated components for VSAM clusters
NOASSOC	for LISTC, does not add associated VSAM components
CHECKCAT	for LISTF, checks the catalog status of each data set as it is added
NOCHECK	default for LISTF, does not check catalog status
MIGRAT	default for LISTC, searches for all data sets
NOMIGRAT	for LISTC, searches for non-migrated data sets
ONLYMIGR	for LISTC, searches for only migrated (MIGRAT or ARCIVE) data sets
OPTICAL	opens the VTOC for Optical devices (3395 M151)
NOOPTICAL	default, does not open the VTOC for optical devices (3395 M151)
QUICK	for LISTC, bypasses data set volume verification
NOQUICK	default for LISTC, verifies data set volume status
RESET	resets the FILTER options to default values.

FILTER Function

Remarks Some special considerations apply for the FILTER command:

1. Migrated data sets are identified by a MIGRAT or ARCIVE volume name.
2. DSORG type checking can not be performed on migrated data sets.
3. If QUICK is active, only GDG, NOVSAM and VSAM can be checked;
   P/PS/PO/PE/PDS/DA/IS all result in non-VSAM data set selection.
4. MIGRAT/NOMIGRAT/ONLYMIGR, QUICK and ASSOC are only applicable to LISTC.
5. CHECKCAT/NOCHECK is applicable to LISTF only.



```
------------------------------ FILTER command  -------------------------------
 OPTION  ===>

Enter any operands below for FILTER:

Note: the following are used by both LISTC and LISTF
 Include by Type  ===> ALL (ALL/P/PS/PO/PE/PDS/DA/IS/NOVS/VS/GDG; data type)
 Access 3395 optical > NO  (Yes/No; Yes allows VTOC access on 3395 M151)

Note: the following is only used by LISTC
 Quick            ===> NO  (Yes/No; Yes bypasses data set volume verification)
 Associations     ===> YES (Yes/No; No bypasses VSAM components)
 Include migrated ===> YES (Yes/No/Only; Yes to include migrated data sets,
                                         Only to only include migrated dsns)

Note: the following is only used by LISTF
 Check catalog    ===> NO  (Yes/No; Read catalog for each data set for match)
```

Figure 44. Sample FILTER Prompt Panel

FIND Subcommand

Purpose The FIND subcommand displays portions of a member which contain a search string. Optionally, the THEN or ELSE keywords may be entered to allow conditional execution of other subcommands. Note that when a THEN keyword is entered, the results of the FIND subcommand are not displayed unless you also enter the DISPLAY keyword.

Example FIND mema:memb   'this data'

Syntax

```
FIND  memgroup
      'string'
      [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP                    ]
      [CAPS/ASIS/IGNORE/PICTURE     WORD/PREFIX/SUFFIX           ]
      [OFFSET(Hx) MODULE({* / Fullm / Partm*)}                   ]
      [AFTER(num) DO(num) MAXMEMBERS(num) COLS(from:to)          ]
      [AND('string2') ACOLS(from:to) ACAPS/AASIS/AIGNORE/APICTURE]
      [OR('string3') OCOLS(from:to) OCAPS/OASIS/OIGNORE/OPICTURE ]
      [FORMAT(from:to,from:to, ...)                              ]
      [SKIPREC(n) MAXIN(n) MAXOUT(n) MAXFIND(n)                  ]
      [SKIPCOL(n) MAXLEN(n) FIRST/NOFIRST                        ]
      [DISPLAY                                                   ]
      [THEN(ATTRIB  / BROWSE  / DIRENTRY / DELETE  / EDIT   /
            EXCLUDE / FSE     / HISTORY  / LIST    / MAP    /
            MEMBERS / MEMLIST / NEWML    / PRINT   / REVIEW /
            SUBMIT  / SUBLIST / TSOEDIT  / TSOLIST / VERIFY / VIEW)  ]
      [ELSE(ATTRIB  / BROWSE  / DIRENTRY / DELETE  / EDIT   /
            EXCLUDE / FSE     / HISTORY  / LIST    / MAP    /
            MEMBERS / MEMLIST / NEWML    / PRINT   / REVIEW /
            SUBMIT  / SUBLIST / TSOEDIT  / TSOLIST / VERIFY / VIEW)  ]

    * [FROMKEY(key)/FROMADDRESS(add)/FROMNUMBER(num)             ]
    * [TOKEY(key)/TOADDRESS(add)/TONUMBER(num)                   ]
*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases FI, FIN, FIND

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults memgroup, previous string, IGNORE, NUM or previous LIST/FIND/REPLACE format

Required none

Operands

memgroup identifies the member(s) to be searched.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

string identifies the data to be located. It is coded as a delimited string like **+searched for+**. If the string is not entered or is null, the last string entered will be used. The string may contain one to 32 characters. If the ASIS keyword is entered, the string will not be translated to upper case letters.

As an alternative, hexadecimal data may be entered as a string delimited with x's like **x0123456789abcdefx.** Note that the string may contain one to 64 characters and that x333x and x0333x are equivalent.

NUM examine the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise) of each logical record for numerics; if the line number field is not numeric, switch formatting to NONUM mode for the remainder of the member.

For ISPF-saved members, the high-order six digits of the line number field is formatted; otherwise, the low-order six digits of the line number field is formatted by suppressing leading zeroes. The line number segment is followed by a blank and up to 249 characters of data from the logical record.

SNUM discard the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise). Search or display up to 256 bytes from a logical record.

NONUM search or display up to 256 bytes from a logical record (without regard to line numbers).

LBLOCK format logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record) and segments of up to 64 characters surrounded by asterisks. Only formatted segments which contain the search string are displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for load modules, only CSECT data will be searched or displayed and if a name is entered in a MODULE operand, only those CSECT or ENTRY names which satisfy the name will be searched or displayed. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset) and segments of up to 64 characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

LDUMP format logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks. Only formatted segments which contain the search string are displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for load modules, only CSECT data will be searched or displayed and if a name is entered in a MODULE operand, only those CSECT or ENTRY names which satisfy the name will be searched or displayed. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset), segments of up to 32 characters of hexadecimal data and up to 16 bytes of characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

BLOCK format physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record) and segments of up to 64 characters surrounded by asterisks.

Only formatted segments which contain the search string are displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

If control interval processing is being performed, the first field is always the control interval relative byte address.

DUMP format physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks.

Only formatted segments which contain the search string are displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

If control interval processing is being performed, the first field is always the control interval relative byte address.

CAPS upper-case any character string search argument.

ASIS do not upper-case the search string.

IGNORE search for upper and lower-case string arguments.

PICTURE if specified, search for data using matching characters and the following picture characters:

=	for any character
%	for any alphanumeric character
@	for any alphabetic character
#	or any numeric character
$	for any special character
~	for any non-blank character (note: the “not” character can also be used)
.	for any any invalid character
-	for any any non-numeric character
<	for any lower case alphabetic character
>	for any upper case alphabetic character

WORD search for strings preceded and followed by a non-alphameric character. This can be used to limit the string hits.

PREFIX search for strings preceded by a non-alphameric character. This can be used to limit the string hits.

SUFFIX search for strings followed by a non-alphameric character. This can be used to limit the string hits.

OFFSET(Hx) specifies a 1 to 6 digit hexadecimal module offset at which the search is to begin. Note: this operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT or ENTRY names to be searched or displayed. Note: this operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected. The MODULE operand has several valid forms:

MODULE(*)	use the previous name entered on any MODULE keyword.
MODULE(Fullm)	search or display only a CSECT or ENTRY named FULLM.
MODULE(Partm*)	search or display only a CSECT or ENTRY named PARTM...

AFTER(n) is coded as 0 through 9999999 to specify the number of matches to skip before reporting FIND strings.

MAXMEMBERS(n) n is coded as 0 through 9999999 to specify the maximum number of members to be selected by FIND.

DO(n) is coded as 0 through 9999999 to specify the maximum number of lines to output for a member.

COLS(from:to) column range to search for the primary string (COLS is only supported for formats NUM, SNUM and NONUM). For example, **COLS(1:10)** specifies that the string may begin in columns 1 through 10.

AND('str2') another string to search in addition to the primary string. AND strings are only supported for formats NUM, SNUM or NONUM and a FIND will be reported for a primary string only if the AND string is found first.

ACOLS(from:to) column range to search for the AND string. For example, **ACOLS(1:10)** specifies that the string may begin in columns 1 through 10.

ACAPS if an AND character string argument is used, it is to be translated to upper case letters.

AASIS if an AND character string argument is used, it is not to be translated to upper case letters.

AIGNORE if an AND character string argument is used, search for upper and lower-case data matching the string.

APICTURE if an AND character string argument is used, search for data with matching characters and the following picture characters:

=	for any character
%	for any alphanumeric character
@	for any alphabetic character
#	or any numeric character
$	for any special character
~	for any non-blank character (note: the “not” character can also be used)
.	for any any invalid character
-	for any any non-numeric character
<	for any lower case alphabetic character
>	for any upper case alphabetic character

OR('str3') another string to search as an alternate to the primary string. OR strings are only supported for formats NUM, SNUM or NONUM and a FIND will be reported for either a primary string or an OR string.

OCOLS(from:to) column range to search for the OR string. For example, **OCOLS(1:10)** specifies that the string may begin in columns 1 through 10.

OCAPS if an OR character string argument is used, it is to be translated to upper case letters.

OASIS if an OR character string argument is used, it is not to be translated to upper case letters.

OIGNORE if an OR character string argument is used, search for upper and lower-case data matching the string.

OPICTURE if an OR character string argument is used, search for data with matching characters and the following picture characters:

=	for any character
%	for any alphanumeric character
@	for any alphabetic character
#	or any numeric character
$	for any special character
~	for any non-blank character (note: the “not” character can also be used)
.	for any any invalid character
-	for any any non-numeric character
<	for any lower case alphabetic character
>	for any upper case alphabetic character

FORMAT(from:to,from:to, ...) specifies the record columns to display in the output when reporting a FIND string. FORMAT is only supported with NUM, SNUM or NONUM data. For example, **FORMAT(21:30,0,1:10)** specifies that the output should be formatted with data from columns 21 through 30, a blank and data from columns 1 through 10.

SKIPREC(n) ignore **n** (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) at the beginning of a member.

MAXIN(n) input up to **n** (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) for a member after satisfying any SKIPREC operand.

MAXOUT(n) display up to **n** (coded as 0 through 9999999) output lines for a member.

MAXFIND(n) locate up to **n** (coded as 0 through 9999999) strings in a member.

SKIPCOL(n) ignore **n** (coded as 0 through 99999) columns at the beginning of each logical record (physical record for BLOCK or DUMP formats).

Note: for NUM or SNUM output format with record format V, SKIPCOL(0) refers to the first data position after the line number field.

MAXLEN(n) search or display up to **n** (coded as 0 through 99999) characters in a logical record (physical record for BLOCK or DUMP formats).

FIRST list all remaining data in the member after a string is located.

NOFIRST list only string matches.

DISPLAY Specifies that the results of the FIND subcommand are to be displayed before taking any THEN action. This allows you to see the lines containing the string and take a conditional action.

THEN(action) execute a subcommand if the string is found in a member. The subcommand may be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY or VIEW. Note that NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET.

ELSE(action) execute a subcommand if the string is not found in a member. The subcommand may be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY or VIEW. Note that NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET.

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access begins at the first record whose key matches (or is greater than) the portion of the key specified. This parameter may be used with TOKEY and it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can be specified for any VSAM data set component if control interval processing is being used.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.

This parameter may be used with TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified.

This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can be specified for any VSAM data set component if control interval processing is being used.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.

This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

Remarks Unprintable characters for 3270-type devices are translated to periods before they are displayed.

FIND formats are NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK and DUMP. For load modules, the NUM, SNUM and NONUM formats are equivalent to the LDUMP format and for VSAM data sets, the NUM, SNUM and NONUM formats are equivalent to the LBLOCK format.

The default format is initially NUM; however, each time a format operand is entered on a LIST, FIND or REPLACE subcommand, the value entered is used as the output format for subsequent LIST, FIND and REPLACE subcommands. Also, note the following:

1. Formats NUM, SNUM and NONUM limit the search and display length for logical records to 256 characters.
2. Column 72 is not searched for formats NUM or SNUM if the record format is fixed with 80 character records as this is normally the "continuation" column.
3. Formats BLOCK and DUMP apply to physical records for non-VSAM data sets (or when control interval processing is being performed); the other formats apply to logical records. For load modules, LBLOCK and LDUMP formats display only CSECT data.
4. Formats LBLOCK, LDUMP, BLOCK and DUMP display only those segments of a record which contain the search string; however, a following segment will also be displayed if the string spans a segment boundary.
5. For VSAM DATA or INDEX components, the LIST, FIND and REPLACE subcommands support control interval access using the DUMP or BLOCK display formats. Instead of accessing individual VSAM records, each GET or PUT obtains a VSAM control interval.

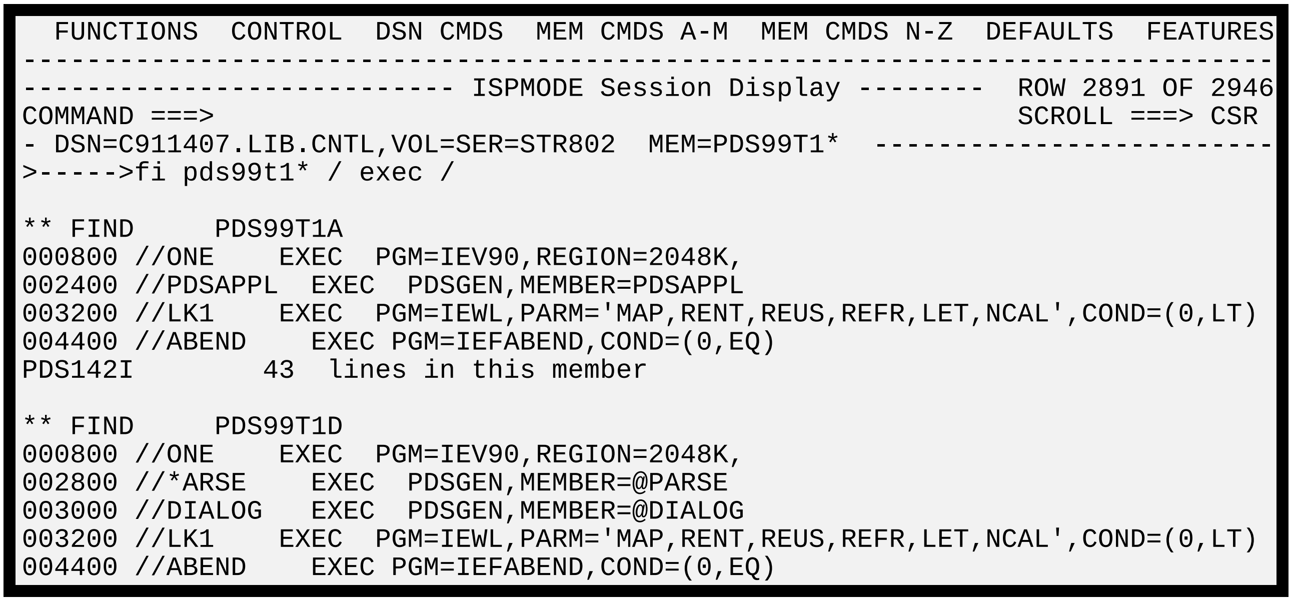
Control interval access could be useful if a VSAM data set has logical errors. REPLACE could be used to repair the error; however, since only the component is opened for update, the next access of the data set through the related cluster will get warning errors due to the differing time stamps.

The various FIND keywords are summarized by type below.

FIND Keyword Summary Table

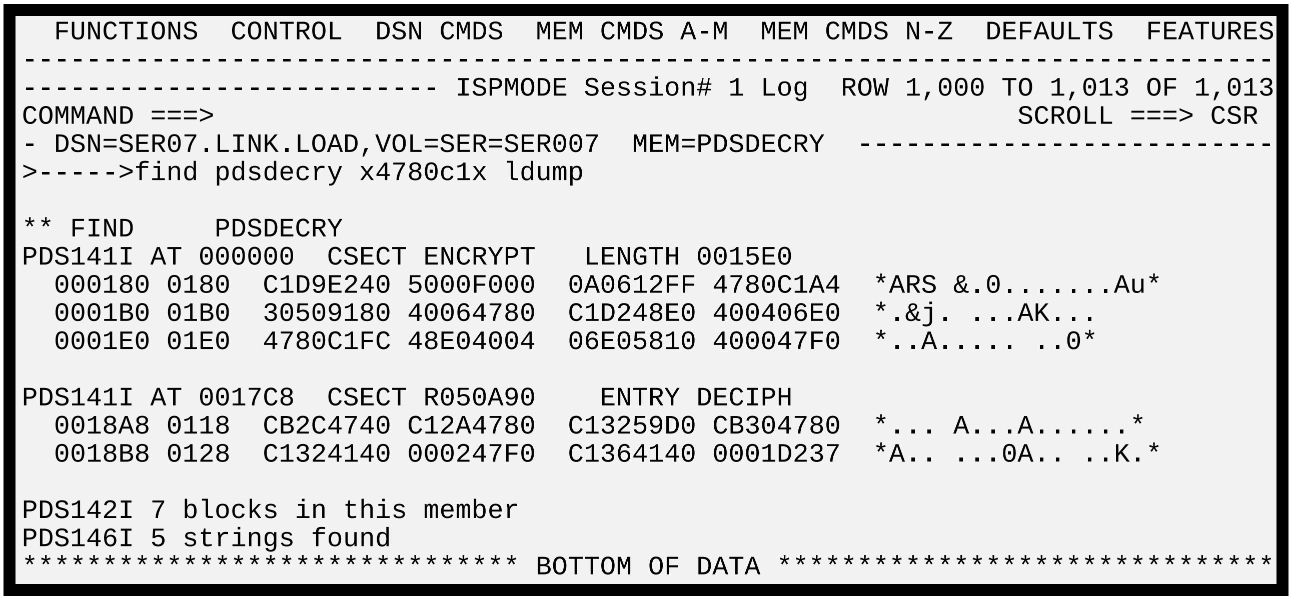
Keyword Type	**Keywords**
format	NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK, DUMP
translate	CAPS, ASIS, IGNORE, PICTURE
position	OFFSET, MODULE, SKIPREC, MAXIN, MAXOUT, MAXFIND, SKIPCOL, MAXLEN
condition	THEN, ELSE
for source members	NUM, SNUM, NONUM, AND, ACOL, OR, OCOL, FORMAT
for load members	OFFSET, MODULE
for VSAM	FROMKEY, FROMADDRESS, FROMNUMBER, TOKEY, TOADDRESS, TONUMBER

FIND Subcommand



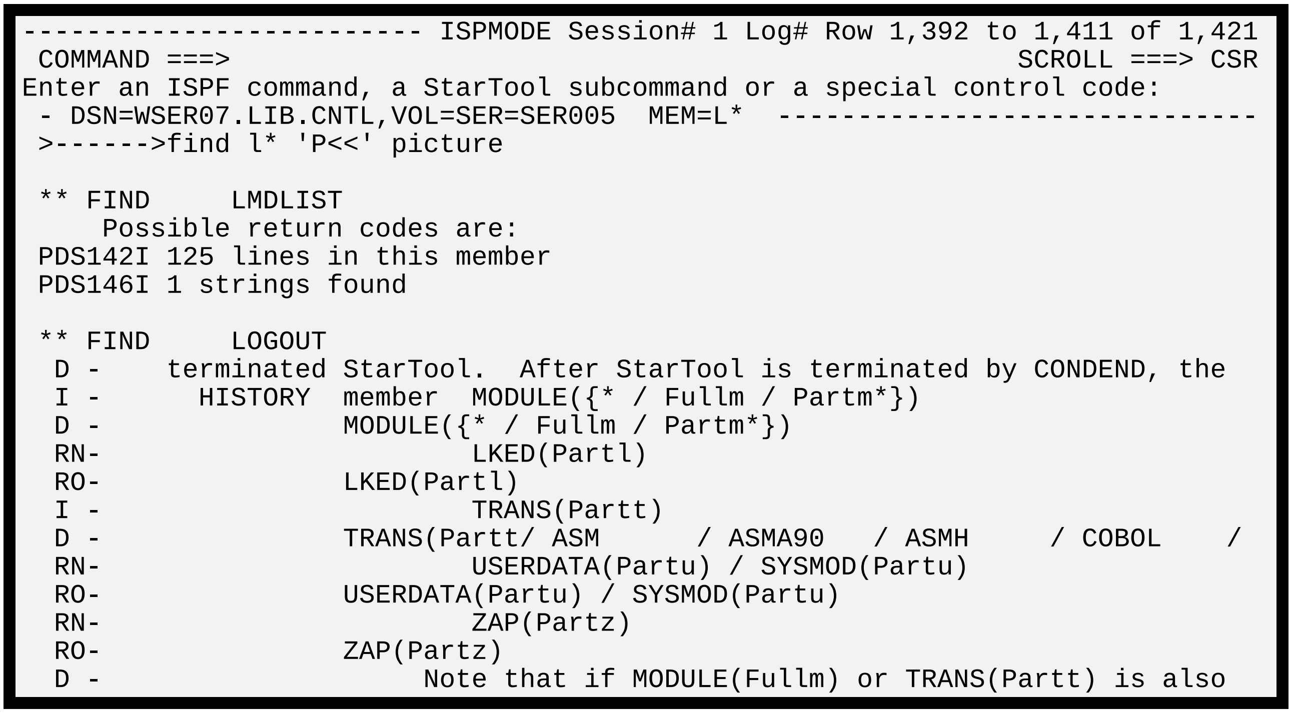
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display --------  ROW 2891 OF 2946
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR802  MEM=PDS99T1*  -------------------------
>----->fi pds99t1* / exec /

** FIND     PDS99T1A
000800 //ONE    EXEC  PGM=IEV90,REGION=2048K,
002400 //PDSAPPL  EXEC  PDSGEN,MEMBER=PDSAPPL
003200 //LK1    EXEC  PGM=IEWL,PARM='MAP,RENT,REUS,REFR,LET,NCAL',COND=(0,LT)
004400 //ABEND    EXEC PGM=IEFABEND,COND=(0,EQ)
PDS142I        43  lines in this member

** FIND     PDS99T1D
000800 //ONE    EXEC  PGM=IEV90,REGION=2048K,
002800 //*ARSE    EXEC  PDSGEN,MEMBER=@PARSE
003000 //DIALOG   EXEC  PDSGEN,MEMBER=@DIALOG
003200 //LK1    EXEC  PGM=IEWL,PARM='MAP,RENT,REUS,REFR,LET,NCAL',COND=(0,LT)
004400 //ABEND    EXEC PGM=IEFABEND,COND=(0,EQ)
```

Figure 45. Sample FIND Subcommand (source)



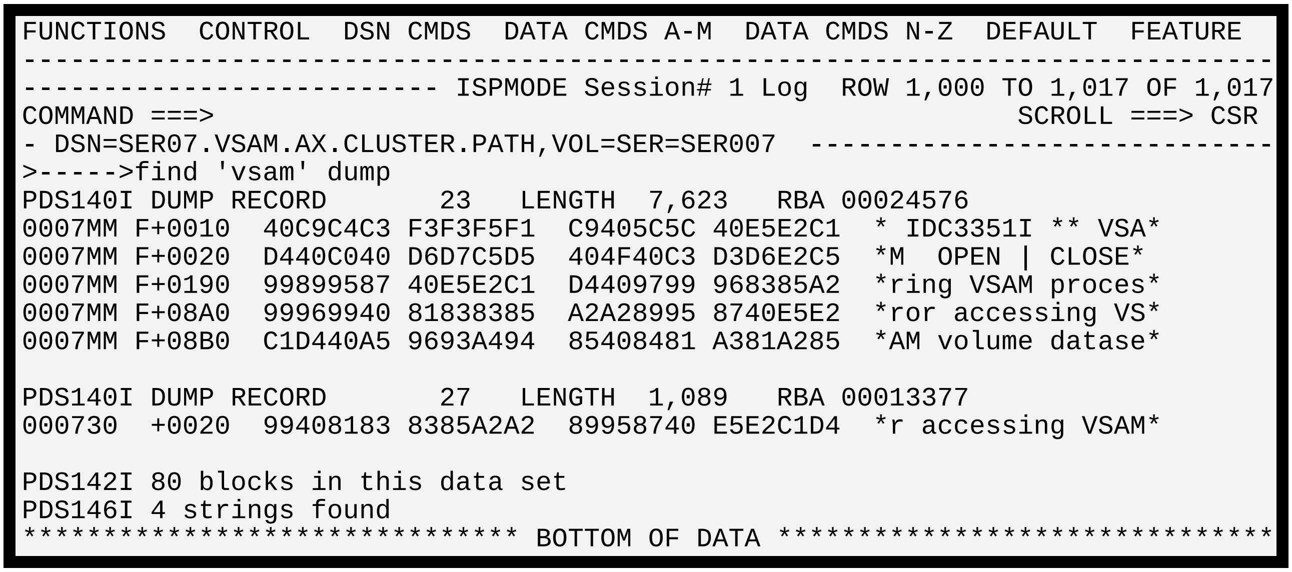
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,013 OF 1,013
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=PDSDECRY  --------------------------
>----->find pdsdecry x4780c1x ldump

** FIND     PDSDECRY
PDS141I AT 000000  CSECT ENCRYPT   LENGTH 0015E0
  000180 0180  C1D9E240 5000F000  0A0612FF 4780C1A4  *ARS &.0.......Au*
  0001B0 01B0  30509180 40064780  C1D248E0 400406E0  *.&j. ...AK...
  0001E0 01E0  4780C1FC 48E04004  06E05810 400047F0  *..A..... ..0*

PDS141I AT 0017C8  CSECT R050A90    ENTRY DECIPH
  0018A8 0118  CB2C4740 C12A4780  C13259D0 CB304780  *... A...A......*
  0018B8 0128  C1324140 000247F0  C1364140 0001D237  *A.. ...0A.. ..K.*

PDS142I 7 blocks in this member
PDS146I 5 strings found
****************************** BOTTOM OF DATA *******************************
```

Figure 46. Sample FIND Subcommand (load)



```
------------------------- ISPMODE Session# 1 Log# Row 1,392 to 1,411 of 1,421
 COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=L*  ------------------------------
 >------>find l* 'P<<' picture

 ** FIND     LMDLIST
    Possible return codes are:
 PDS142I 125 lines in this member
 PDS146I 1 strings found

 ** FIND     LOGOUT
  D -    terminated StarTool.  After StarTool is terminated by CONDEND, the
  I -      HISTORY  member  MODULE({* / Fullm / Partm*})
  D -               MODULE({* / Fullm / Partm*})
  RN-                       LKED(Partl)
  RO-               LKED(Partl)
  I -                       TRANS(Partt)
  D -               TRANS(Partt/ ASM      / ASMA90   / ASMH     / COBOL    /
  RN-                       USERDATA(Partu) / SYSMOD(Partu)
  RO-               USERDATA(Partu) / SYSMOD(Partu)
  RN-                       ZAP(Partz)
  RO-               ZAP(Partz)
  D -                    Note that if MODULE(Fullm) or TRANS(Partt) is also
```

Figure 47. Sample FIND Subcommand with a picture



```
FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,017 OF 1,017
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.VSAM.AX.CLUSTER.PATH,VOL=SER=SER007  -----------------------------
>----->find 'vsam' dump
PDS140I DUMP RECORD       23   LENGTH  7,623   RBA 00024576
0007MM F+0010  40C9C4C3 F3F3F5F1  C9405C5C 40E5E2C1  * IDC3351I ** VSA*
0007MM F+0020  D440C040 D6D7C5D5  404F40C3 D3D6E2C5  *M  OPEN | CLOSE*
0007MM F+0190  99899587 40E5E2C1  D4409799 968385A2  *ring VSAM proces*
0007MM F+08A0  99969940 81838385  A2A28995 8740E5E2  *ror accessing VS*
0007MM F+08B0  C1D440A5 9693A494  85408481 A381A285  *AM volume datase*

PDS140I DUMP RECORD       27   LENGTH  1,089   RBA 00013377
000730 +0020   99408183 8385A2A2  89958740 E5E2C1D4  *r accessing VSAM*

PDS142I 80 blocks in this data set
PDS146I 4 strings found
******************************* BOTTOM OF DATA *******************************
```

Figure 48. Sample FIND Subcommand (VSAM Path)

FINDMOD Subcommand

Purpose The FINDMOD subcommand searches for all copies of a system routine.

Example FINDMOD iefbr14

Syntax

```
FINDMOD module [SYSTEM/NUCLEUS/ADDRESS]
               [CHANGE/GO  LIB(num)   ]
               [NOSEARCH              ]
```

Aliases FINDM, FINDMO, FINDMOD

Defaults SYSTEM, LIB(1)

Required module

Operands

module identifies the member to be found.

SYSTEM A BLDL is to be issued and the LPA and MLPA are to be searched. If the member is in MLPA, TASKLIB or linklist, the linklist concatenation is to be searched; if the member is in LPA, the LPALIB concatenation is to be searched. Also, the in-storage nucleus is to be searched.

NUCLEUS the search should take place in the in-storage nucleus.

ADDRESS module is an address whose corresponding routine is to be located in the LPA, MLPA or NUCLEUS. The linklist libraries are to be searched if the module is in MLPA and the LPALIB concatenation is to be searched if it is in LPA.

CHANGE specifies that if the member is found, a CHANGE to that library is to be performed.

GO specifies that if the member is found, a GO session for that library is to be initiated.

LIB(n) if there are multiple finds, LIB specifies which library should be selected for CHANGE or GO. If you specify a higher number for LIB than actually found, the last data set will be selected.

NOSEARCH specifies that the linklist and LPALIB library concatenations are not to be searched.

Remarks If FINDMOD is entered with the SYSTEM parameter (the default) the following search takes place:

1. A BLDL is issued and reported on.
2. The LPA and MLPA is searched (the module may be in both).
3. If the module is in MLPA, TASKLIB or the linklist, the linklist libraries are individually searched.
4. If in LPA, the LPALIB libraries are individually searched.
5. The in-storage nucleus map is searched.

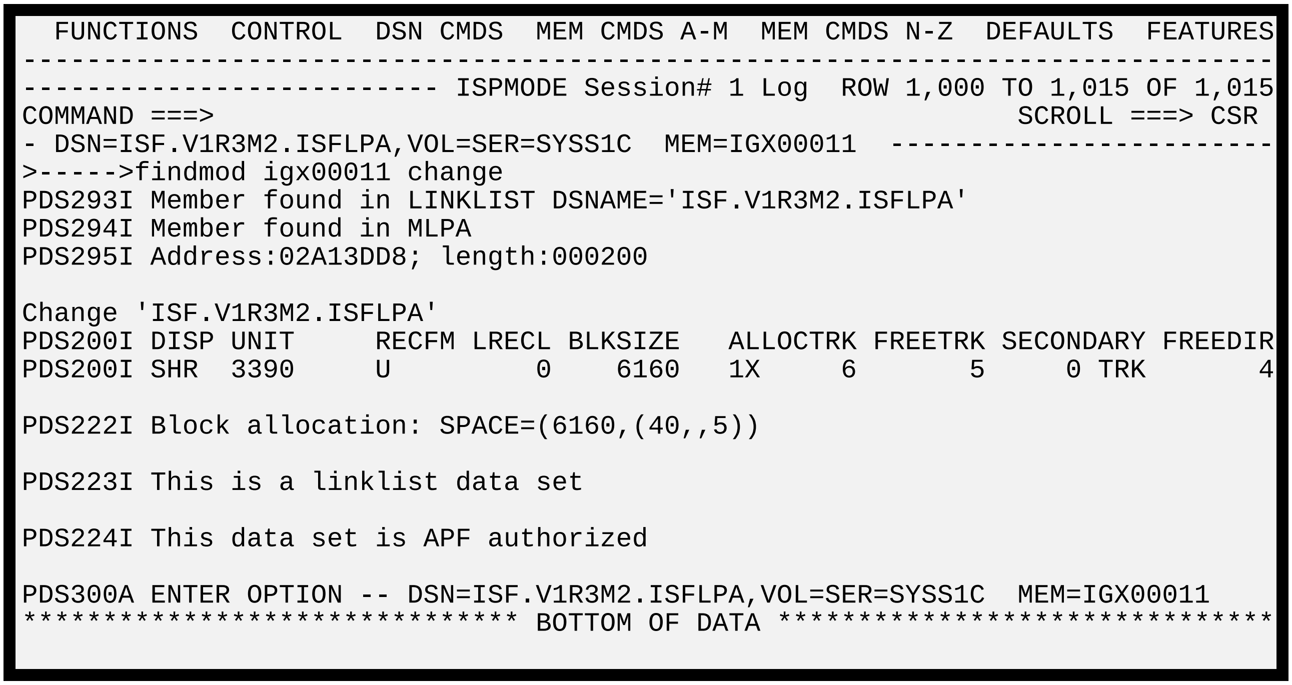
If FINDMOD is entered with the ADDRESS parameter, the module parameter is assumed to be a hex address and the following search takes place:

1. The in-storage nucleus map is searched.

2. If in MLPA, the linklist libraries are individually searched.
3. If in LPA, the LPALIB libraries are individually searched.

If FINDMOD is entered with the NUCLEUS parameter, only the in-storage nucleus map is searched.

FINDMOD supports both dynamic and static LPA and LINKLIST.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,015 OF 1,015
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=ISF.V1R3M2.ISFLPA,VOL=SER=SYSS1C  MEM=IGX00011  ------------------------
>----->findmod igx00011 change
PDS293I Member found in LINKLIST DSNAME='ISF.V1R3M2.ISFLPA'
PDS294I Member found in MLPA
PDS295I Address:02A13DD8; length:000200

Change 'ISF.V1R3M2.ISFLPA'
PDS200I DISP UNIT     RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3390     U         0    6160   1X     6       5     0 TRK       4

PDS222I Block allocation: SPACE=(6160,(40,,5))

PDS223I This is a linklist data set

PDS224I This data set is APF authorized

PDS300A ENTER OPTION -- DSN=ISF.V1R3M2.ISFLPA,VOL=SER=SYSS1C  MEM=IGX00011
******************************* BOTTOM OF DATA *******************************
```

Figure 49. Sample FINDMOD Subcommand

FSE Subcommand

Purpose The FSE subcommand edits a member. The TSO FSE command is used.

Example FSE mema:memb cntl

Syntax

```
FSE   memgroup [ASIS
               [OLD/NEW
               [NONUM
               [ASM/BASIC/CLIST/CNTL/COBOL/DATA/
                FORTE/FORTG/FORTGI/FORTH/GOFORT/
                IPLI/LIST/PLI/PLIF/TEXT/VSBASIC ]
```

Aliases FS, FSE

Defaults memgroup, EDIT type based on the data set name

Required none

Operands

memgroup identifies the member(s) to be edited.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

ASIS	edit the member with upper and lower case characters.
OLD	verify that the member exists before calling FSE.
NEW	verify that the member does not exist before calling FSE.
NONUM	edit the member without using line numbers.
ASM	EDIT type is assembly
BASIC	EDIT type is BASIC
CLIST	EDIT type is CLIST
CNTL	EDIT type is CNTL
COBOL	EDIT type is COBOL
DATA	EDIT type is DATA
FORTE	EDIT type is FORTRAN level "E"
FORTG	EDIT type is FORTRAN level "G"
FORTGI	EDIT type is FORTRAN level "GI"
FORTH	EDIT type is FORTRAN level "H"
GOFORT	EDIT type is GOFORT
IPLI	EDIT type is IPLI
LIST	EDIT type is LIST
PLI	EDIT type is PLI
PLIF	EDIT type is PLI (F-level)
TEXT	EDIT type is TEXT
VSBASIC	EDIT type is VSBASIC

Remarks If an EDIT type keyword is entered, that keyword is passed to FSE as the descriptive qualifier; otherwise, a descriptive qualifier is chosen as follows:

1. If the low-level qualifier of the partitioned data set name is one of the valid descriptive qualifiers for FSE (ASM, BASIC, CLIST, CNTL, COBOL, DATA, IPLI, LIST, PLI, TEXT or VSBASIC), that qualifier is passed to FSE.
2. If the low-level qualifier is FORT, GOFORT is passed to FSE.
3. Otherwise, the data type qualifier is not a valid descriptive qualifier and the general descriptive qualifier, DATA, is passed to FSE.

This interface is optional, it should only be used if you have FSE+ installed at your installation.

GO Function

Purpose The GO command allows you to initiate or switch to parallel STARWARP sessions.

Example GO 'sys1.parmlib'

Syntax

```
GO [* / ** / gonum / dsname [VOLUME(volser)] [ SHR/OLD]
      /  FILE(ddname) [NUMBER(num)]
   [REPLACE                         ]
```

Aliases G, GO

Defaults SHR or previously used GO data set if ***** is entered.

Required none

Operands

***** Switches to the most recently used GO session. This can be used to toggle GO sessions; this is similar to the toggling for **CHANGE ***

****** Rotates through active GO sessions; you might consider setting a PF key to **GO ****

gonum Specifies the number (1 through 9) of the parallel session to initiate or switch to.

dsname Specifies the data set name for the parallel session to initiate or switch to. If the data set name is not entered in quotes ('), your TSO PREFIX will be appended to the start of the entered data set name.

VOLUME(volser) Specifies the volume name to use for uncataloged data sets.

SHR Allocates the data set with a disposition of SHR; allows simultaneous use of this data set by others. The use of **SHR** is recommended.

OLD Allocates the data set with a disposition of OLD; does not allow simultaneous use of this data set by others. The use of **SHR** is recommended.

FILE(ddname) Identifies the DDNAME of a preallocated data set. Note that only disk data sets (including VIO) are supported. If the FILE keyword is used, dsname, SHR/OLD, VOLSET and VOLUME should not also be used. However, if the data set is concatenated, the SHR or OLD keyword may be used since STARWARP reallocates the data set.

NUMBER(num) Specifies the concatenation number desired for the DDNAME allocation for the FILE keyword. Note that num defaults to 1 but if num is larger than the number of concatenated data sets, the last data set in the sequence will be used.

REPLACE Replaces a session (this is for use if a session number and a data set name are both specified).

Remarks The initial session invoked by STARWARP is called GO session 1; if you wish to establish an alternate parallel session, enter GO and a session number or a data set name. You will be prompted for the data set name if necessary and a parallel session will be initiated. To transfer to an established parallel session, enter GO and the session number or data set name.

When you are in an alternate session, note that the log is shared between sessions. MEMLIST and CSECTS processing is maintained uniquely between the different sessions and the other functions are shared between sessions.

Note that a GO session can be deleted with the DROP command. The syntax is:
DROP [*/num] [PROMPT]

HELP Subcommand

Purpose

The HELP subcommand provides information on using STARWARP. HELP operates differently depending on your operating mode. For example, if you are in ISPMODE or MEMLIST, HELP provides standard ISPF tutorial information. In line mode or batch, the HELP subcommand lists data from the HELP data set.

Example

HELP

Syntax

```
HELP   [subcommand [FUNCTION] [SYNTAX] [OPERAND/OPERAND(k)]]
        Note: the above operands are only defined in line mode or batch.
```

Aliases

H, HE, HEL, HELP

Defaults

none

Required

none

Operands

subcommand specifies the name of a subcommand (or the alias name of a subcommand) for which HELP is to be displayed. If this operand is omitted, a list of valid subcommands is displayed.

FUNCTION specifies that function information for the subcommand is to be displayed.

SYNTAX specifies that syntax information for the subcommand is to be displayed.

OPERAND(k) specifies that information for the keyword identified in the (k) operand is to be displayed. If the (k) operand is omitted, information on all operands of the subcommand will be displayed.

Remarks

The HELP subcommand provides information on using STARWARP.

Either the TSO HELP or HEL command is used; this choice was made during STARWARP installation. To determine which program is used, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "HELP calls".

This interface is optional, it should only be used with the HELP command processor or the public domain HEL command.

HEX Function

Purpose The HEX command performs hexadecimal calculations.

Example HEX

Syntax

```
HEX
```

Aliases HEX

Operands (no operands are supported for the HEX subcommand).

Remarks The HEX ISPF table is displayed in response to a HEX command. When you are in a HEX display, you have many options: you may delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as a primary command.

The HEX table displays the calculator tape for your calculator. You can enter new calculations from the HEX display.

The following primary commands are supported directly for the HEX function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

EDITT[BL] (or **ET[BL]**) enters an edit session on HEX table data.

F finds a string and positions the display start location.
Syntax: `F anystring [ASIS]`
`[FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]`

OUT[PUT] outputs the HEX table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the HEX table based on a string match.
Syntax: `REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]`

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

X clears the HEX table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the HEX table; this is equivalent to **X ALL**

HISTORY Subcommand

Purpose The HISTORY subcommand displays the last linkage edit date for a module and lists any CSECT IDR data assigned to a load module.

Example HISTORY   mema:memb

Syntax

```
HISTORY memgroup
       [COBOL(ADV/NOADV                 APOST/NOAPOST
              AWO/NOAWO                 CMPR2/NOCMPR2
              COUNT/NOCOUNT             COMPCLASS/COMPPROGRAM
              CURRENCY/NOCURRENCY       DATA24/NODATA24
              DATA31/NODATA31           DATEPROC/NODATEPROC
              DBCS/NODBCS               DECK/NODECK
              DUMP/NODUMP               DYNAM/NODYNAM
              ENDJOB/NOENDJOB           EVENTS/NOEVENTS
              FASTSRT/NOFASTSRT         FDUMP/NOFDUMP
              INTDATELIL/INTDATEANSI    FLOW/NOFLOW
              LIB/NOLIB                 LIST/NOLIST
              MAP/NOMAP                 NAME/NONAME
              NUMBER/NONUMBER           NUMCLSALT/NUMCLSPRIM
              NUMPROCMIG/NONUMPROCMIG   NUMPROCPFD/NONUMPROCPFD
              OBJECT/NOOBJECT           OBJ370/NOOBJ370
              OFFSET/NOOFFSET           OPTFULL/NOOPT
              OPTIMIZE/NOOPTIMIZE       OUTDD/NOOUTDD
              PGMLONGMIX/NOPGMLONGMIX   PGMLONGUPP/NOPGMLONGUPP
              QUOTE/NOQUOTE             READYTRACE/NOREADYTRACE
              RENT/NORENT               RESIDENT/NORESIDENT
              RMODEANY/RMODE24          SEQUENCE/NOSEQUENCE
              SIZEMAX/NOSIZEMAX         SOURCE/NOSOURCE
              SSRANGE/NOSSRANGE         STATE/NOSTATE
              SYMDMP/NOSYMDMP           TERMINAL/NOTERMINAL
              TEST/NOTEST               TESTBLOCK/NOTESTBLOCK
              TESTPATH/NOTESTPATH       TESTSTMT/NOTESTSTMT
              TRUNCBIN/TRUNCBIN         TRUNCSTD/NOTRUNCSTD
              VBREF/NOVBREF             WORD/NOWORD
              XREF/NOXREF               ZWB/NOZWB )             ]
       [GENERATE                                                ]
       [EXTERN/WKEXTERN/PSEUDOREG/LABELREF/COMMON/PRIVATE/CSECT ]
       [MODULE({* / Fullm / Partm*})                            ]
       [LKED(Partl) / NOLKED                                    ]
       [TRANS(Partt/ ASM      / ASMA90   /  ASMH     / COBOL    /
                     COBOLII  / COBOLVS  /  COBOLV4  / COBOL370 /
                     C370     / FORTRAN  /  FORTRANG / FORTRANH /
                     FORTRANV / IFOX     /  PASCAL   / PLS      /
                     PLX370   / PL1      /  REX370   / RPG      /
                     SASC     )  / NOTRANS                      ]
        [USERDATA(Partu) / SYSMOD(Partu) / NOUSERDATA / NOSYSMOD]
        [ZAP(Partz) / NOZAP                                     ]
        [ALL                                                    ]
        [NOSYSTEM                                               ]
        [MEMBERS / MEMLIST / ML / NEWML / SUBLIST               ]
```

Aliases HI, HIS, HIST, HISTO, HISTOR, HISTORY

Defaults memgroup, TRANS/NOTRANS from CONTROL, LKED, USERDATA, ZAP; but if LKED, USERDATA, ZAP or TRANS are specified, only the corresponding types of data will be provided.

Required none

Operands

memgroup identifies the load member(s) for which history data is desired.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

COBOL(options) specifies one or more COBOL compiler options that are to be matched for a COBOL CSECT. If all options entered for the given compiler type match for a CSECT, the CSECT will be listed.

Note that if MODULE(Fullm) or TRANS(Partt) is also entered, the COBOL keyword applies only to those CSECTS which pass that filtering.

The following COBOL VS and COBOL V4 compiler options are listed alphabetically. Unless otherwise noted, each option applies to both compilers:

COUNT/NOCOUNT	(COBOL VS) Verb count summary
ENDJOB/NOENDJOB	Free main storage at ENDJOB
FLOW/NOFLOW	Flow trace output for abend
OBJ370/NOOBJ370	OBJECT COMPUTER is OBJ370
OPTIMIZE/NOOPTIMIZE	Optimize object code
READYTRACE/NOREADYTRACE	(COBOL VS) READY TRACE
RESIDENT/NORESIDENT	Library Management to load
STATE/NOSTATE	Statement number for abend
SYMDMP/NOSYMDMP	Formatted dump for an abend
TEST/NOTEST	Object code for debug

Notes:

1. The following COBOL VS and COBOL V4 options can affect program execution: COUNT (COBOL VS only) FLOW, OBJ370 OPTIMIZE, RESIDENT, STATE and TEST.
2. All five COBOL compilers use OPT/NOOPT and TEST/NOTEST options.
3. COBOL II and earlier COBOL compilers support the RESIDENT/NORESIDENT option (COBOL for MVS & VM and COBOL for OS/390 & VM set the RESIDENT bit on)

The following COBOL II, COBOL for MVS & VM, and COBOL for OS/390 & VM compiler options are listed alphabetically. Unless otherwise noted, each option applies to all three compilers:

ADV/NOADV	Byte for printer control character
APOST/NOAPOST	Apostrophe (') is non-numeric delimiter
AWO/NOAWO	APPLY WRITE-ONLY clause
CMPR2/NOCMPR2	COBOL II release 2 compatible code
COMPCLASS/COMPPROGRAM	(not COBOL II) Compile unit is class
CURRENCY/NOCURRENCY	(not COBOL II) Alternate symbol for $
DATA24/NODATA24	Dynamic storage is below 16 Megabytes
DATA31/NODATA31	Dynamic storage is unrestricted
DATEPROC/NODATEPROC	(not COBOL II) Unknown at present
DBCS/NODBCS	Double Byte Character Set shift codes
DECK/NODECK	Object code output to //SYSPUNCH
DUMP/NODUMP	System dump if the compiler abends
DYNAM/NODYNAM	CALLed programs invoked dynamically

EVENTS/NOEVENTS	(not COBOL II) Messages to SYSEVENT
FASTSRT/NOFASTSRT	Invoke external product to perform sort
FDUMP/NOFDUMP	(COBOL II) Formatted dump for abend
INTDATELIL/INTDATEANSI	(not COBOL II) Lilian for date basis
LIB/NOLIB	Process COPY, BASIS & REPLACE
LIST/NOLIST	Compiler listing of generated code
MAP/NOMAP	DATA DIVISION map
NAME/NONAME	Link-edit NAME for each object module
NUMBER/NONUMBER	Line numbers processed in columns 1 to 6
NUMCLSALT/NONUMCLSPRIM	(not COBOL II) Numeric class test
NUMPROCMIG/NONUMPROCMI	Invalid sign processing for migrate
NUMPROCPFD/NONUMPROCPFD	Invalid sign processing for speed
OBJECT/NOOBJECT	Object code output to //SYSLIN
OFFSET/NOOFFSET	Condensed PROCEDURE DIVISION list
OPTFULL/NOOPT	(not COBOL II) OPTIMIZE(FULL)
OPTIMIZE/NOOPTIMIZE	Optimize object code
OUTDD/NOOUTDD	DISPLAY ddname instead of //SYSOUT
PGMLONGMIX/NOPGMLONGMIX	(not COBOL II) Names are ASIS
PGMLONGUPP/NOPGMLONGUPP	(not COBOL II) Not truncated
QUOTE/NOQUOTE	Quote mark (") is non-numeric delimiter
RENT/NORENT	Reentrant object code
RESIDENT/NORESIDENT	(COBOL II) Library Management to load
RMODEANY/RMODE24	(not COBOL II) RMODE(ANY)
SEQUENCE/NOSEQUENCE	Sequence numbers processed
SIZEMAX/NOSIZEMAX	SIZE(MAX) used in compilation
SOURCE/NOSOURCE	Source listing
SSRANGE/NOSSRANGE	Check subscripts, indexes and references
TERMINAL/NOTERMINAL	Diagnostic messages to //SYSTERM
TEST/NOTEST	Object code for debug
TESTBLOCK/NOTESTBLOCK	(not COBOL II) TEST(BLOCK)
TESTPATH/NOTESTPATH	(not COBOL II) TEST(PATH)
TESTSTMT/NOTESTSTMT	(not COBOL II) TEST(STMT)
TRUNCBIN/NOTRUNCBIN	Binary field truncation protection
TRUNCOPT/NOTRUNCOPT	Binary field truncation for performance
VBREF/NOVBREF	Verb cross reference
WORD/NOWORD	Alternate reserved word list
XREF/NOXREF	Symbol cross reference for names
ZWB/NOZWB	Remove sign in compare DISPLAY fields

Notes:

1. The following COBOL options can affect program execution: ADV, AWO, CMPR2, DATA, DYNAM, FASTSRT, FDUMP (COBOL II only), OPTIMIZE, OUTDD, NUMPROC, RENT, RESIDENT (COBOL II only), SSRANGE, TEST, TRUNC and ZWB.
2. All COBOL compilers have OPT/NOOPT and TEST/NOTEST options.
3. COBOL II and earlier COBOL compilers support the RESIDENT/NORESIDENT option (COBOL for MVS & VM and COBOL for OS/390 & VM set the RESIDENT bit on)

GENERATE specifies that output from the HISTORY subcommand is to be filtered and reformatted into a form which can be processed by a program or a rexx exec as described briefly below:

1. The "** HIST" message contains linkage edit attributes for the module.
2. The PDS250I message contains all bit settings for COBOL VS and V4
3. The PDS251I message TYP field contains **ASM**, **PL1**, **FOR**, **RPG**, **REX**, **C37**, **MAP** or **???** with the third character overlaid with **D** (for DB2), **I** (for IMS), **O** (for ONLINE), **B** (for DB2 and ONLINE), or **S** (for Secondary if the CSECT name is not the same as the module name).
4. The PDS255I message contains all run time option bit settings for COBOL II, COBOL for MVS & VM, or COBOL for OS/390 & VM.
5. The PDS260I message contains all bit settings for COBOL II, COBOL for MVS & VM or COBOL for OS/390 & VM and the TYP field will be **II**, **MVS** or **OS** with the third character overlaid with **D** (for DB2), **I** (for IMS), **O** (for ONLINE), **B** (for DB2 and ONLINE), or **S** (for Secondary if the CSECT name is not the same as the module name).
6. Most other HISTORY messages are suppressed.

The format of each message with GENERATE is defined in the PDS250I, PDS251I, PDS255I and PDS260I message section. The ****HIST** message separator line is changed as shown below:

**** HIST COBOLTST** Linkage attributes
Col 9: CSECT name
Col 23: **RENT** or blank
Col 28: **REUS** or blank
Col 33: **REFR** or blank
Col 38: **OVLY** or blank
Col 43: **TEST** or blank
Col 48: **SCTR** (scatter loaded) or blank
Col 53: **OL** (only loadable) or blank
Col 56: **NE** (not editable) or blank
Col 59: **RANY** or **R24**
Col 64: **AANY**, **A31** or **A24**
Col 69: **RL=** followed by the real member name for an alias

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT names for history reporting. Note that if this parameter is entered with any TRANS SYSMOD/USERDATA or ZAP keywords, only those CSECTs which

EXTERN specifies that load modules with missing external symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

WKEXTERN specifies that load modules with missing weak symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

PSEUDOREG specifies that load modules with PSEUDO register symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

LABELREF specifies that load modules with ENTRY symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

COMMON specifies that load modules with COMMON area symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

PRIVATE specifies that load modules with PRIVATE area symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

CSECT specifies that load modules with CSECT symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT names for history reporting. Note that if this parameter is entered with any TRANS

SYSMOD/USERDATA or ZAP keywords, only those CSECTs which satisfy the MODULE keyword are checked for the other keyword conditions.

The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.

MODULE(Fullm) report only on CSECT FULLM.

MODULE(Partm*) report only on CSECT PARTM...

LKED(partl) specifies that if the linkage editor name matches the partial name, **Partl**, a line identifying the editor is to be listed. **Partl** may be entered as a partial linkage editor name with one to ten characters.

NOLKED specifies that the HISTORY information listed should not include linkage editor information. Other types of HISTORY output are not affected.

SYSMOD(Partu) specifies that user IDR data which matches the partial name, **Partu**, is to be listed. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the SYSMOD operand applies only to CSECTs which satisfy the MODULE operand.

NOSYSMOD specifies that the HISTORY information listed should not include SYSMOD information. Other types of HISTORY output are not affected.

TRANS(Partt) specifies that CSECTS identified as having been assembled or compiled by the translator defined by the partial name, **Partt**, are to be listed. **Partt** may be entered as a partial translator name with one to ten characters. Note that if MODULE(...) is also entered, the TRANS operand applies only to CSECTs which satisfy the MODULE operand.

If a generic translator name from below is entered instead of Partt, the translator types shown below will be used instead.

ASM IFOX, IEV90 or ASMA90 assembler (Translators 52ASM32B88, 52ASM31686, 566896201, 5734AS1, 569623400, 5741SC103, 360SAS037, 360SAS038 or 5734AS100)

ASMA90 High level assembler (Translator 569623400)

ASMH Assembler H (Translators 566896201 and 5734AS1)

COBOL COBOL V4, COBOL VS, COBOL II or COBOL 370 (Translators 566895801, 566895807, 566895901, 40CB1, 5740CB103, 5752SC104, 40CB-1, 12345 or 12345-1).

COBOLII COBOL II (Translators 566895801 or 566895901)

COBOLMVS COBOL for MVS & VM (Translator 566895807)

COBOLOS COBOL for OS/390 & VM (Translator 5648A2500)

COBOLVS COBOL VS (Translators 5740CB103, 40CB1, 40CB-1, 12345 or 12345-1).

COBOLV4 COBOL V4 (Translators 5752SC104, 12345 or 12345-1).

C370 C/370 (Translators 5688187, 5688216 or 5688040)

FORTRAN VS FORTRAN or FORTRAN G or H (Translators 5668-806, 5734-FO2, 5734-FO3, 5734-F02, 5748-FO3 or 5796-PKR)

FORTRANG FORTRAN G (Translators 5734-FO2 or 5734-F02)

FORTRANH FORTRAN H (Translator 5734-FO3)

FORTRANV VS FORTRAN (Translators 5668-806, 5748-FO3 or 5796-PKR)

IFOX VS assembler (Translators 52ASM32B88, 52ASM31686, 15741SC103, 360SAS037, 360SAS038 or 5734AS100)

PASCAL VS PASCAL (Translator 566876701)

PLS PLS-III (Translator C'PLS-III' or CA-PLNK 0)

PLX370	PL/X 370 (Translator PL/X-370)
PL1	PL/I (Translators 5734-PL1, 5668-910 or 5688-235)
REXX370	Compiled rexx (Translator 569501301)
RPG	RPG (Translator 5740RG1)
SASC	SAS/C (Translators BLD121988, LC370B, SAS/C, SAS/C/ or SDS080888)

NOTRANS specifies that the HISTORY information listed should not include TRANSLATOR information. Other types of HISTORY output are not affected.

USERDATA(Partu) specifies that user IDR data which matches the partial name, **Partu**, is to be listed. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the USERDATA operand applies only to CSECTs which satisfy the MODULE operand.

NOUSERDATA specifies that the HISTORY information listed should not include USERDATA information. Other types of HISTORY output are not affected.

ZAP(Partz) specifies that zap IDR data which matches the partial name, **Partz**, is to be listed. **Partz** may be entered as partial zap IDR data with one to eight characters. Note that if MODULE(...) is also entered, the ZAP operand applies only to CSECTs which satisfy the MODULE operand.

NOZAP specifies that the HISTORY information listed should not include ZAP information. Other types of HISTORY output are not affected.

ALL specifies that LKED, TRANS, USERDATA, ZAP is the reporting default. This option is often used with one of the negative options as in the following example: **HISTORY name NOTRANS ALL**

NOSYSTEM specifies that system modules are to be filtered out before reporting MAP information as is done for HISTORY GENERATE. System or compiler routines begin with DFH, DFS, DSN, IBM, IEY, IGY, ILB, ISP, or PLI.

MEMBERS displays the names of members which satisfy the HISTORY subcommand without changing the current member group.

MEMLIST Same as **ML**. Specifies that any member displayed by the HISTORY subcommand will be selected for MEMLIST display. The MODULE(name), LKED(Partl), TRANS(Partt), USERDATA(Partu) or SYSMOD(Partu) and ZAP(Partz) keywords are used to search for members to display. If no members are selected, a null sublist is the result.

ML Same as **MEMLIST**. Specifies that any member displayed by the HISTORY subcommand will be selected for MEMLIST display. The MODULE(name), LKED(Partl), TRANS(Partt), USERDATA(Partu) or SYSMD(Partu) and ZAP(Partz) keywords are used to search for members to display. If no members are selected, a null sublist is the result.

NEWML Same as **MEMLIST** and **ML** except that the current MEMLIST is reset.

SUBLIST Specifies that any member displayed by the HISTORY subcommand will be selected for inclusion in a new sublist. The MODULE(name), LKED(Partl), TRANS(Partt), USERDATA(Partu) or SYSMOD(Partu) and ZAP(Partz) keywords are used to search for members to display. If no members are selected, a null sublist is the result.

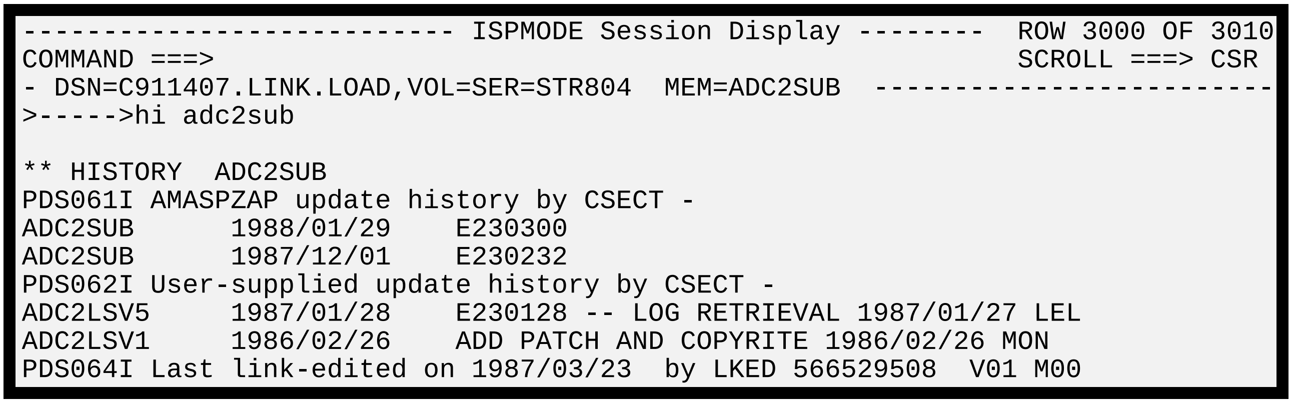
Remarks If any CSECTS in a load member were compiled by a COBOL compiler, several compile time options will be listed for each COBOL CSECT (see messages PDS250I and PDS260I for more information).

Note: if a member is not a load module, any ISPF statistics or any SSI information will be displayed for the member and any HISTORY keyword information will be ignored.

HISTORY Subcommand

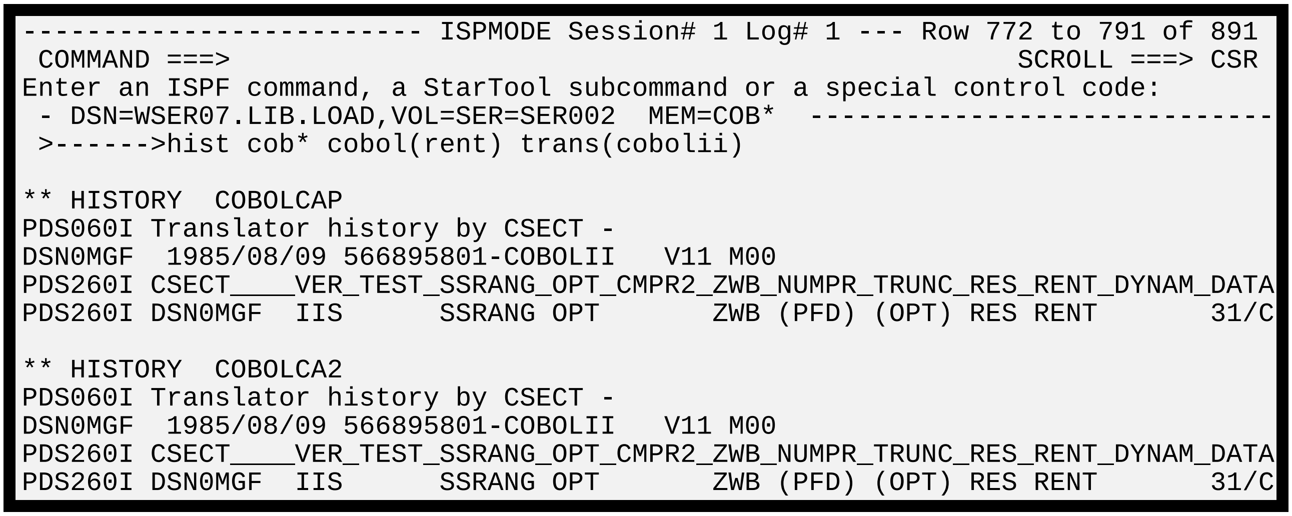
If DFSMS APAR OW29593 is applied on your system, the binder records hours, minutes and seconds as well as the date in the history IDR record. The StarTool PDS064I message was changed to report this information in HH:MM format if it is present:

PDS064I Last link-edited on 1999/06/18 by 5695DF108-BINDER ...
PDS064I Last link-edited on 1999/06/23 14:04 by 5695DF108-BINDER ...



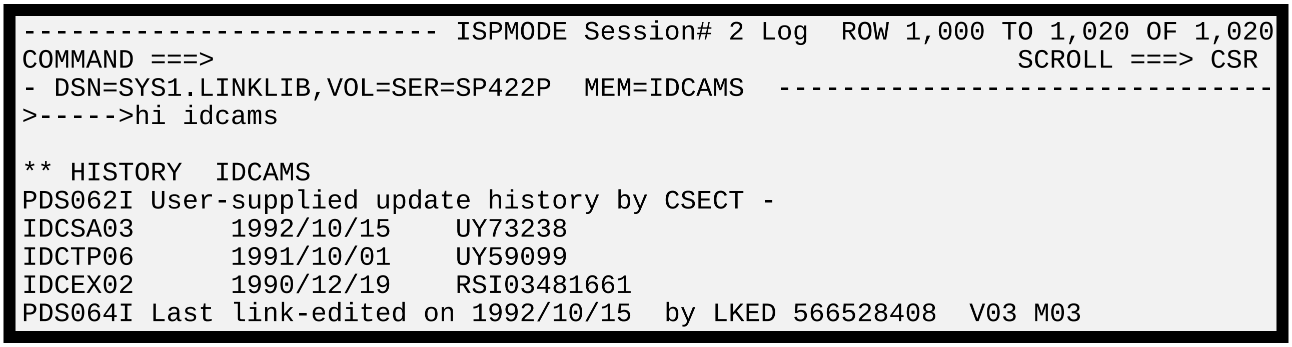
```
-------------------------- ISPMODE Session Display --------  ROW 3000 OF 3010
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ADC2SUB  ------------------------
>----->hi adc2sub

** HISTORY  ADC2SUB
PDS061I AMASPZAP update history by CSECT -
ADC2SUB      1988/01/29    E230300
ADC2SUB      1987/12/01    E230232
PDS062I User-supplied update history by CSECT -
ADC2LSV5     1987/01/28    E230128 -- LOG RETRIEVAL 1987/01/27 LEL
ADC2LSV1     1986/02/26    ADD PATCH AND COPYRITE 1986/02/26 MON
PDS064I Last link-edited on 1987/03/23  by LKED 566529508  V01 M00
```

Figure 52. Sample HISTORY Subcommand



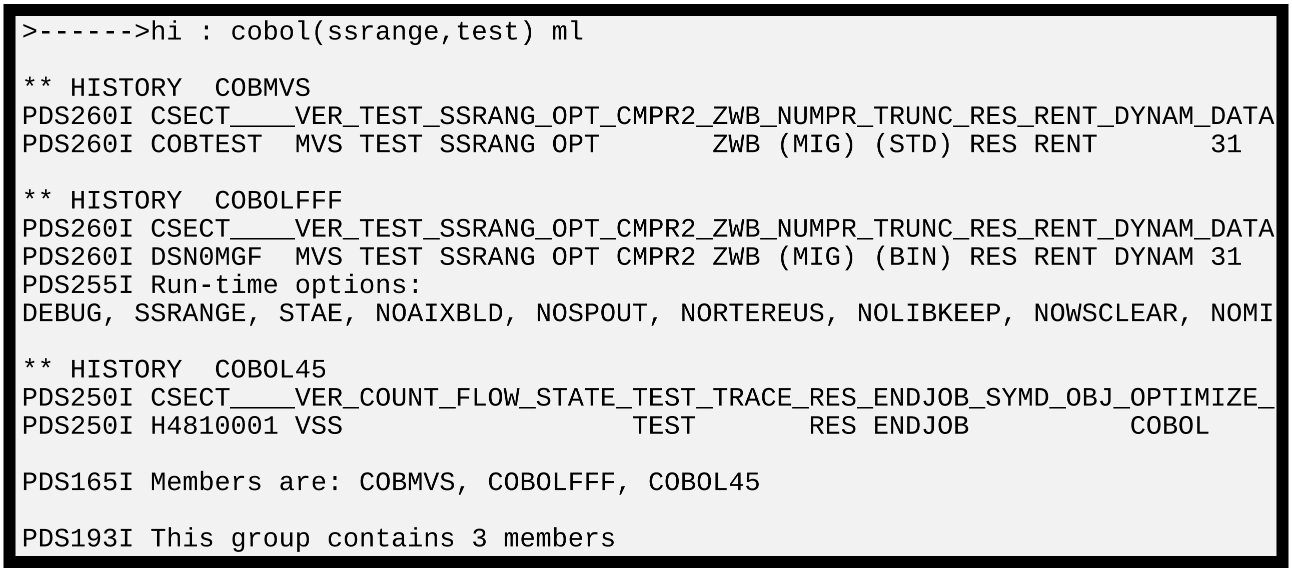
```
------------------------ ISPMODE Session# 1 Log# 1 --- Row 772 to 791 of 891
 COMMAND ===>                                                 SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.LOAD,VOL=SER=SER002  MEM=COB*  ----------------------------
 >------>hist cob* cobol(rent) trans(cobolii)

** HISTORY  COBOLCAP
PDS060I Translator history by CSECT -
DSN0MGF  1985/08/09 566895801-COBOLII   V11 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF  IIS      SSRANG OPT       ZWB (PFD) (OPT) RES RENT       31/C

** HISTORY  COBOLCA2
PDS060I Translator history by CSECT -
DSN0MGF  1985/08/09 566895801-COBOLII   V11 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF  IIS      SSRANG OPT       ZWB (PFD) (OPT) RES RENT       31/C
```

Figure 53. Sample HISTORY Search for COBOL(RENT)



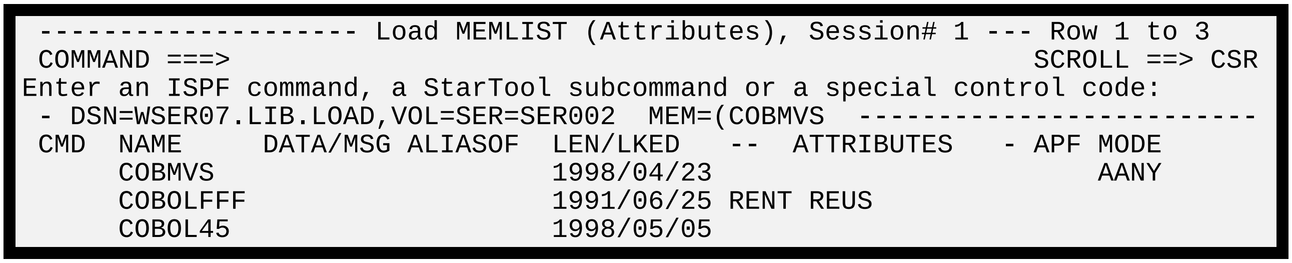
```
------------------------- ISPMODE Session# 2 Log  ROW 1,000 TO 1,020 OF 1,020
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SYS1.LINKLIB,VOL=SER=SP422P  MEM=IDCAMS  ------------------------------
>----->hi idcams

** HISTORY  IDCAMS
PDS062I User-supplied update history by CSECT -
IDCSA03      1992/10/15    UY73238
IDCTP06      1991/10/01    UY59099
IDCEX02      1990/12/19    RSI03481661
PDS064I Last link-edited on 1992/10/15  by LKED 566528408  V03 M03
```

Figure 54. Sample HISTORY Subcommand with IBM data



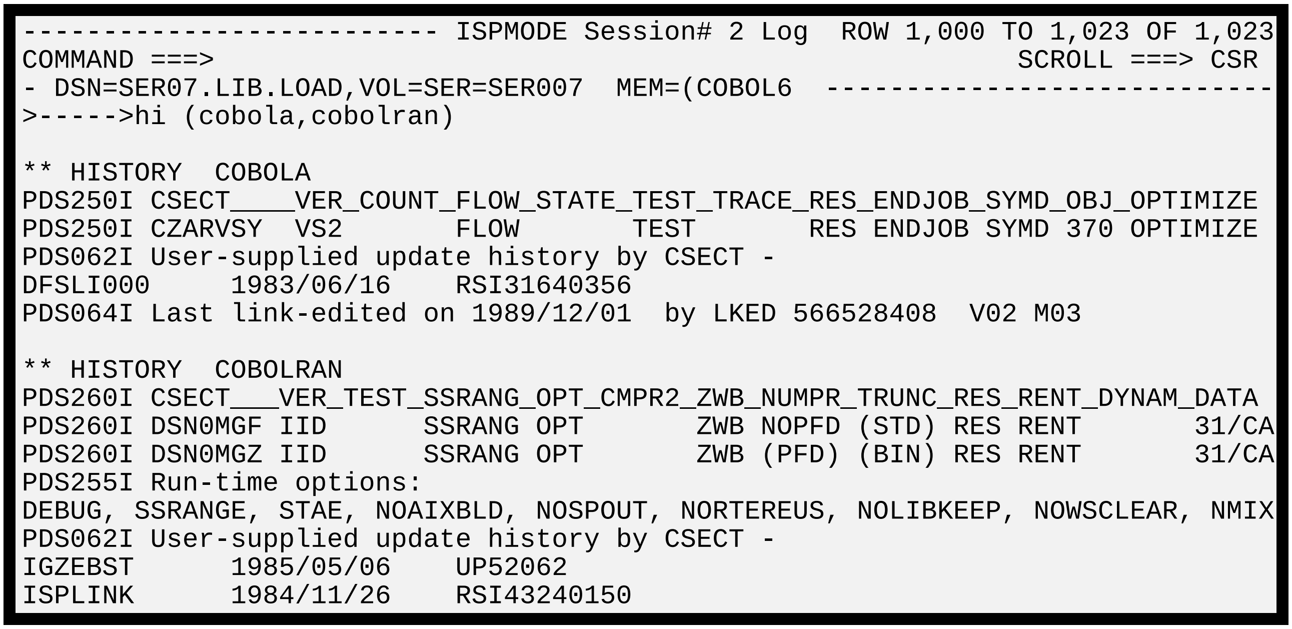
```
>------>hi : cobol(ssrange,test) ml

** HISTORY  COBMVS
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I COBTEST  MVS TEST SSRANG OPT       ZWB (MIG) (STD) RES RENT       31

** HISTORY  COBOLFFF
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF  MVS TEST SSRANG OPT CMPR2 ZWB (MIG) (BIN) RES RENT DYNAM 31
PDS255I Run-time options:
DEBUG, SSRANGE, STAE, NOAIXBLD, NOSPOUT, NORTEREUS, NOLIBKEEP, NOWSCLEAR, NOMI

** HISTORY  COBOL45
PDS250I CSECT____VER_COUNT_FLOW_STATE_TEST_TRACE_RES_ENDJOB_SYMD_OBJ_OPTIMIZE_
PDS250I H4810001 VSS                  TEST       RES ENDJOB          COBOL

PDS165I Members are: COBMVS, COBOLFFF, COBOL45

PDS193I This group contains 3 members
```

Figure 54. Sample HISTORY Subcommand to search for COBOL options



```
 -------------------- Load MEMLIST (Attributes), Session# 1 --- Row 1 to 3
 COMMAND ===>                                                  SCROLL ==> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.LIB.LOAD,VOL=SER=SER002  MEM=(COBMVS  -------------------------
 CMD  NAME     DATA/MSG ALIASOF  LEN/LKED   --  ATTRIBUTES   - APF MODE
      COBMVS                    1998/04/23                      AANY
      COBOLFFF                  1991/06/25 RENT REUS
      COBOL45                   1998/05/05
```

Figure 54. Sample HISTORY Subcommand resulting member list



```
-------------------------- ISPMODE Session# 2 Log  ROW 1,000 TO 1,023 OF 1,023
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.LOAD,VOL=SER=SER007  MEM=(COBOL6  ----------------------------
>----->hi (cobola,cobolran)

** HISTORY  COBOLA
PDS250I CSECT____VER_COUNT_FLOW_STATE_TEST_TRACE_RES_ENDJOB_SYMD_OBJ_OPTIMIZE
PDS250I CZARVSY  VS2       FLOW       TEST       RES ENDJOB SYMD 370 OPTIMIZE
PDS062I User-supplied update history by CSECT -
DFSLI000     1983/06/16    RSI31640356
PDS064I Last link-edited on 1989/12/01  by LKED 566528408  V02 M03

** HISTORY  COBOLRAN
PDS260I CSECT___VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSN0MGF IID      SSRANG OPT       ZWB NOPFD (STD) RES RENT       31/CA
PDS260I DSN0MGZ IID      SSRANG OPT       ZWB (PFD) (BIN) RES RENT       31/CA
PDS255I Run-time options:
DEBUG, SSRANGE, STAE, NOAIXBLD, NOSPOUT, NORTEREUS, NOLIBKEEP, NOWSCLEAR, NMIX
PDS062I User-supplied update history by CSECT -
IGZEBST      1985/05/06    UP52062
ISPLINK      1984/11/26    RSI43240150
```

Figure 55. Sample HISTORY Subcommand for COBOL

HISTORY Subcommand

```
>------>hi (cobol6,dsx0mgf,cobos390) nosyst

** HISTORY  COBOL6
PDS060I Translator history by CSECT -
HM01A    1985/01/29 40CB1-COBOLVS       V20 M00
DATECALC 1983/06/20 5734AS100-IFOX00    V05 M01
UABEND   1983/06/20 5734AS100-IFOX00    V05 M01
PDS250I CSECT____VER_COUNT_FLOW_STATE_TEST_TRACE_RES_ENDJOB_SYMD_OBJ_OPTIMIZE_
PDS250I HM01A    VSS                                 ENDJOB      370 CAPEX
PDS061I AMASPZAP update history by CSECT -
PDS067I Member has 1 IDR blocks with space for 19 IDR entries
PDS068I 1 IDR entries are in use; 18 are available for use
PDS064I Last link-edited on 1989/12/01 by 566528408-DFPLKED    V02 M03

** HISTORY  DSX0MGF
PDS060I Translator history by CSECT -
DSX0MGF  1985/08/09 566895801-COBOLMVS  V11 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I DSX0MGF  IID      SSRANG OPT       ZWB (PFD) (OPT) RES RENT       31
PDS061I AMASPZAP update history by CSECT -
PDS067I Member has 1 IDR blocks with space for 19 IDR entries
PDS068I 1 IDR entries are in use; 18 are available for use
PDS064I Last link-edited on 1991/06/25 by 566528408-DFPLKED    V03 M01

** HISTORY  COBOS390
PDS060I Translator history by CSECT -
COBTEST  1998/04/23 5648A2500-COBOLOS   V12 M00
PDS260I CSECT____VER_TEST_SSRANG_OPT_CMPR2_ZWB_NUMPR_TRUNC_RES_RENT_DYNAM_DATA
PDS260I COBTEST  OSS                       ZWB NOPFD (STD) RES            31
PDS067I Member has 1 IDR blocks with space for 19 IDR entries
PDS068I 1 IDR entries are in use; 18 are available for use
PDS064I Last link-edited on 1998/04/23 by 5695DF108-BINDER     V01 M01
```

Figure 55. Sample HISTORY Subcommand with NOSYSTEM

```
>------>hi (cobol6,dsx0mgf,cobos390) gen

** HIST COBOL6                                            R24  A24
PDS250I HM01A    VSS  NNNNNNNNNNNNNNYNYYYNNNNN
PDS251I DATECALC ASS
PDS251I UABEND   ASS

** HIST DSX0MGF       RENT REUS                           RANY A31  RL=BOBZIM
PDS260I DSX0MGF  IID  NYYNYNNNNNYNYYYNYYYNYYYYNNNNYYNNNNNNNNNN
PDS255I Run-time options: NNNNNNNNNNYYYNNNNNNNNNNNNNNNNNNNNNNNNNNN

** HIST COBOS390                                          R24  AANY
PDS260I COBTEST  OSS  YNYNNNNNNNNNYNNNNNYYYYNNNYNNNYNNNNNNNNNNNNNNNNNNNNNNNNNN
```

Figure 55. Sample HISTORY Subcommand with GENERATE

IDCAMS Subcommand

Purpose The IDCAMS subcommand invokes IDCAMS passing it a command and operands and displays the output from the command.

Example IDCAMS listcat entry(userid.vsam.ix.cluster) all

Syntax

```
IDCAMS   command  operands
```

Aliases ID, IDC, IDCA, IDCAM, IDCAMS

Defaults none

Required IDCAMS command and operands; if no IDCAMS command is entered in ISPF mode, an IDCAMS prompting sequence is initiated.

Operands

command specifies the IDCAMS command desired: ALLOCATE, ALTER, BLDINDEX, CNVTCAT, DEFINE, DELETE, EXAMINE, EXPORT, IMPORT, LISTCAT, PRINT, REPRO or VERIFY.

IDCAMS is invoked as an APF-authorized processor via the TSO/E IKJEFTSR interface. Thus, operations on a catalog which require authorization (DEFINE with RECATALOG, DEFINE with RECOVERY, EXPORT, IMPORT, PRINT, REPRO or VERIFY) can be performed.

operands specifies any additional operands for the command. For example, DEFINE needs an initial operand of ALIAS, AIX, CLUSTER, GDG, NONVSAM, PAGESPACE, PATH or USERCATALOG followed by additional operands to complete the DEFINE command.

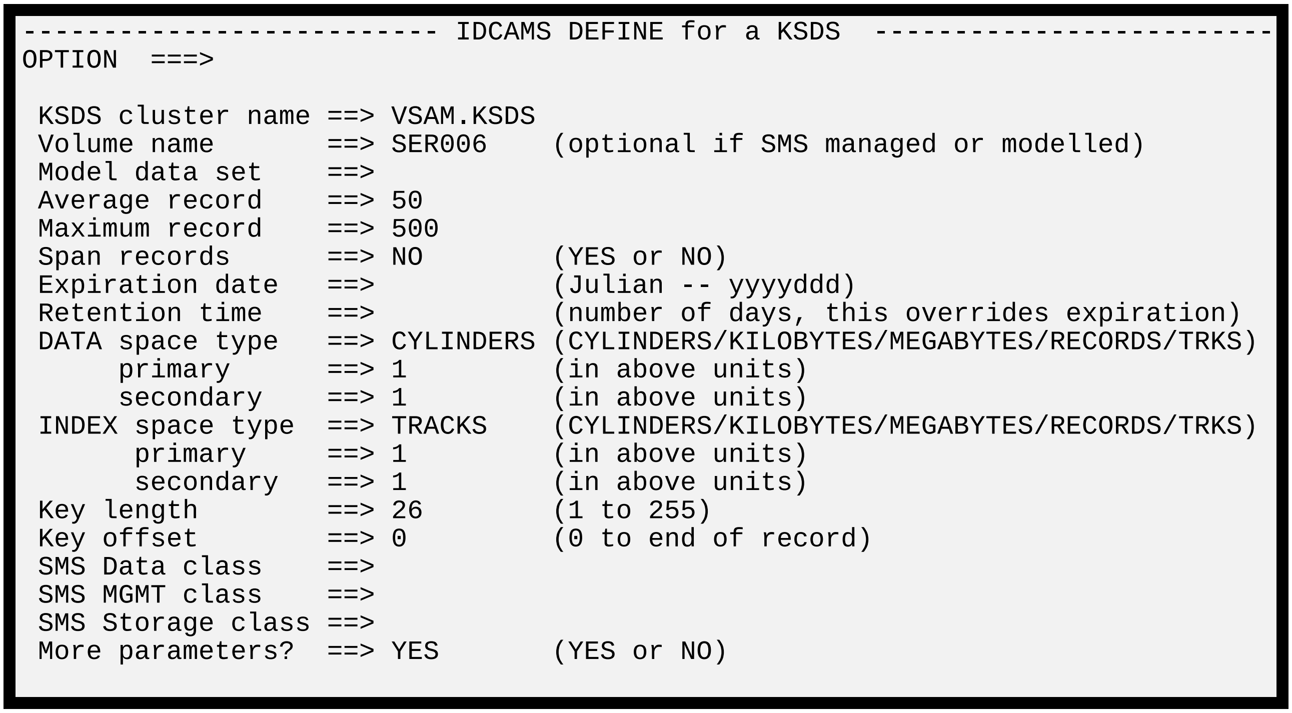
Remarks If you invoke the IDCAMS subcommand with no operands in an ISPF environment, you will be prompted for which of several IDCAMS commands you wish to perform; previous operands entered for each command are remembered across ISPF sessions.

An IDCAMS subcommand issued from STARWARP is limited to a maximum of 245 characters of data. This restriction may be a problem if you are entering IDCAMS DEFINE commands directly; the IDCAMS DEFINE prompts use an alternative interface which allows up to 800 characters.

If the IDCAMS prompting interface is used, normal TSO conventions are used for data set names but catalog names must be fully qualified. When IDCAMS is invoked with a command and operands, however, all data set and catalog names must be fully qualified but you may optionally use quote marks around data set names.

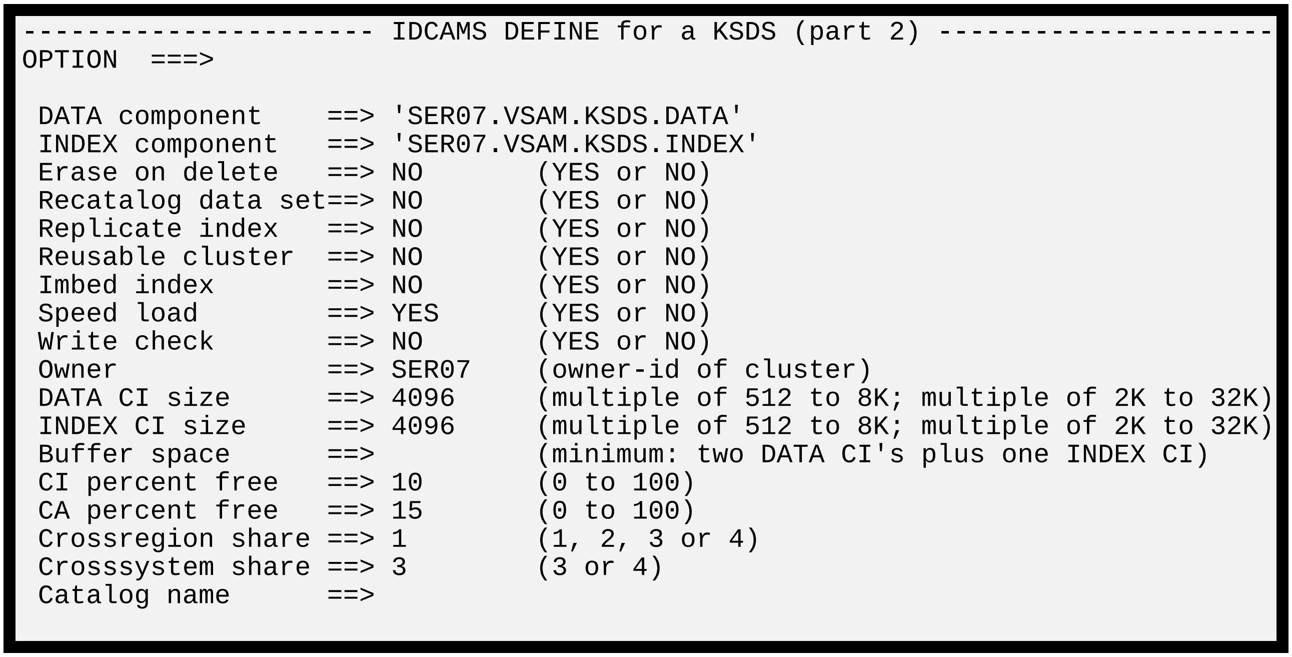
Note: **DEFINE** is a short-hand command for entering IDCAMS with no operands and selecting the DEFINE option.

The following DEFINE KSDS panel may be reached by **IDCAMS;DEFINE;KSDS** or **DEFINE;KSDS**



```
------------------------- IDCAMS DEFINE for a KSDS  ------------------------
OPTION  ===>

 KSDS cluster name ==> VSAM.KSDS
 Volume name       ==> SER006    (optional if SMS managed or modelled)
 Model data set    ==>
 Average record    ==> 50
 Maximum record    ==> 500
 Span records      ==> NO        (YES or NO)
 Expiration date   ==>           (Julian -- yyyyddd)
 Retention time    ==>           (number of days, this overrides expiration)
 DATA space type   ==> CYLINDERS (CYLINDERS/KILOBYTES/MEGABYTES/RECORDS/TRKS)
      primary      ==> 1         (in above units)
      secondary    ==> 1         (in above units)
 INDEX space type  ==> TRACKS    (CYLINDERS/KILOBYTES/MEGABYTES/RECORDS/TRKS)
       primary     ==> 1         (in above units)
       secondary   ==> 1         (in above units)
 Key length        ==> 26        (1 to 255)
 Key offset        ==> 0         (0 to end of record)
 SMS Data class    ==>
 SMS MGMT class    ==>
 SMS Storage class ==>
 More parameters?  ==> YES       (YES or NO)
```

Figure 56. Sample IDCAMS DEFINE (part 1)



```
--------------------- IDCAMS DEFINE for a KSDS (part 2) --------------------
OPTION  ===>

 DATA component    ==> 'SER07.VSAM.KSDS.DATA'
 INDEX component   ==> 'SER07.VSAM.KSDS.INDEX'
 Erase on delete   ==> NO       (YES or NO)
 Recatalog data set==> NO       (YES or NO)
 Replicate index   ==> NO       (YES or NO)
 Reusable cluster  ==> NO       (YES or NO)
 Imbed index       ==> NO       (YES or NO)
 Speed load        ==> YES      (YES or NO)
 Write check       ==> NO       (YES or NO)
 Owner             ==> SER07    (owner-id of cluster)
 DATA CI size      ==> 4096     (multiple of 512 to 8K; multiple of 2K to 32K)
 INDEX CI size     ==> 4096     (multiple of 512 to 8K; multiple of 2K to 32K)
 Buffer space      ==>          (minimum: two DATA CI's plus one INDEX CI)
 CI percent free   ==> 10       (0 to 100)
 CA percent free   ==> 15       (0 to 100)
 Crossregion share ==> 1        (1, 2, 3 or 4)
 Crosssystem share ==> 3        (3 or 4)
 Catalog name      ==>
```

Figure 57. Sample IDCAMS DEFINE (part 2)

IF Subcommand

Purpose The IF subcommand searches for members meeting defined conditions. The action to be taken is specified by the THEN and ELSE keywords. Either the THEN or ELSE keyword may be omitted; if both THEN and ELSE are omitted, a default of THEN(ATTRIB) is assumed.

If all conditions are met for a given member, any THEN action is taken for that member; otherwise, any ELSE action is taken.

Example IF mema:memb changed(93/11/24:93/11/30) then(edit)

Syntax

```
IF memgroup
   [SINCE/BEFORE                                     ]
   [TODAY/YESTERDAY/WEEK/CURRENT/BIWEEK/
          MONTH/QUARTER/HALFYEAR/YEAR/BIYEAR/
          LAST(numdays)/DATE(yyyy/mm/dd)             ]
   [CHANGED(yyyy/mm/dd:yyyy/mm/dd)                   ]
   [CREATED(yyyy/mm/dd:yyyy/mm/dd)                   ]
   [ABOVE(Count1)                                    ]
   [ALIAS/NOALIAS                                    ]
   [AMODE24/AMODE31/AMODEANY/NOAMODE24/
            NOAMODE31/NOAMODEANY                     ] (load only)
 * [APFERR/NOAPFERR                                  ] (load only)
 * [APPARENTALIAS/NOAPPARENTALIAS                    ]
   [AUTH/NOAUTH                                      ] (load only)
   [BELOW(Count2)                                    ]
 * [BLOCKERR/MAXBLK(size)/NOBLOCKERR                 ]
   [DC/NODC                                          ] (load only)
   [EDIT/NOEDIT                                      ] (load only)
   [EXEC/NOEXEC                                      ] (load only)
 * [EXTERN/WKEXTERN/PSEUDOREG/LABELREF/COMMON/PRIVATE/CSECT] (load)
   [FLEVEL/NOFLEVEL                                  ] (load only)
   [HASALIAS/NOHASALIAS                              ]
   [ID(Puid)/NOID/NOTID(Puid)                        ]
 * [IDRFULL/NOIDRFULL                                ] (load only)
 * [IOERR/NOIOERR                                    ]
   [LKED(Partl)                                      ] (load only)
 * [LKEDERR/NOLKEDERR                                ] (load only)
 * [LOADERR/NOLOADERR                                ] (load only)
   [LOADONLY/NOLOADONLY                              ] (load only)
 * [LRECLERR/NOLRECLERR                              ]
   [MODULE({* / Fullm / Partm*)}                     ] (load only)
 * [NAMEERR/NONAMEERR                                ]
   [NULL/NONULL                                      ]
   [ORPHAN/NOORPHAN                                  ]
   [OVERLAY/NOOVERLAY                                ] (load only)
  *[PACKED/NOPACKED                                  ]
   [PAGE/NOPAGE                                      ] (load only)
   [REFR/NOREFR                                      ] (load only)
   [RENT/NORENT                                      ] (load only)
   [REUS/NOREUS                                      ] (load only)
 * [RLDERR/NORLDERR                                  ] (load only)
 * [RLDZERO/NORLDZERO                                ] (load only)
   [RMODE24/RMODEANY/NORMODE24/NORMODEANY            ] (load only)
   [SCTR/NOSCTR                                      ] (load only)
 * [SPFEDIT/NOSPFEDIT                                ]
   [SSI(hexdata)/SSI/NOSSI/PARTSSI(hexdata)          ]
   [SYSMOD(Partu) / USERDATA(Partu)                  ] (load only)
   [TEST/NOTEST                                      ] (load only)
   [TRANS(Partt)                                     ] (load only)
   [TTR(Lttr:Httr)                                   ]
   [USERDATA(Partu) / SYSMOD(Partu)                  ] (load only)
   [USERID(Puid)/NOUSERID/NOTUSERID(Puid)            ]
   [VSLKED/NOVSLKED                                  ] (load only)
   [ZAP(Partz)                                       ] (load only)
 [THEN(ATTRIB / BROWSE  / DIRENTRY / DELETE   / EDIT    /
                EXCLUDE / FIND     / FSE      / HISTORY / LIST   /
                MAP     / MEMBERS  / MEMLIST  / NEWML   / PRINT  /
                REVIEW  / SUBMIT   / SUBLIST  / TSOEDIT / VERIFY /
                VIEW                                 ]
 [ELSE(ATTRIB / BROWSE  / DIRENTRY / DELETE   / EDIT    /
                EXCLUDE / FIND     / FSE      / HISTORY / LIST   /
                MAP     / MEMBERS  / MEMLIST  / NEWML   / PRINT  /
                REVIEW  / SUBMIT   / SUBLIST  / TSOEDIT / VERIFY /
                VIEW                                 ]
* Note: the * marked tests are unique to IF as compared to ML or EXCLUDE
```

Aliases I, IF

Defaults memgroup, SINCE, THEN(ATTRIB) if neither THEN nor ELSE is entered

Required none

Operands

memgroup identifies the member(s) whose attributes are to be examined.

Default member names, member lists, member name ranges and member name patterns are allowed; see **Appendix A. Member Name Forms** on page 261.

SINCE use dates from the specified date to the current date. SINCE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

FROM use dates from the specified date to the current date. FROM indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

BEFORE use dates before (and including) the specified date. BEFORE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

TO use dates before (and including) the specified date. TO indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

TODAY a date, equivalent to LAST(0)

YESTERDAY a date, equivalent to LAST(1)

WEEK a date, equivalent to LAST(7)

CURRENT a date, equivalent to LAST(10)

BIWEEK a date, equivalent to LAST(14)

MONTH a date, equivalent to LAST(30)

QUARTER a date, equivalent to LAST(120)

HALFYEAR a date, equivalent to LAST(183)

YEAR a date, equivalent to LAST(365)

BIYEAR a date, equivalent to LAST(730)

LAST(numdays) a date, indicates the number of days before today.

DATE(cdate) a date, indicates the actual date; it may be entered in ISPF format (yyyy/mm/dd), or Julian format (yyyy.ddd).

CHANGED(date: searches for members modified by an ISPF editor or a ZAP program between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like:
CHANGED(yyyy/mm/dd:yyyy/mm/dd)

Note that only members with ISPF statistics or members created by a linkage editor are supported.

CREATED(date: searches for members created with ISPF statistics or by the linkage editor between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like:
CREATED(yyyy/mm/dd:yyyy/mm/dd)

ABOVE(Count1) select members with more than the number of lines (amount of storage for load modules) defined by the number, Count1.

ALIAS select alias members.

NOALIAS select main members.

AMODE24 select modules with addressing mode 24.

NOAMODE24 select modules with addressing mode 31 or ANY.

AMODE31 select modules with addressing mode 31.

NOAMODE31 select modules with addressing mode 24 or ANY.

AMODEANY select modules with addressing mode ANY.

NOAMODEANY select modules with addressing mode 24 or 31.

APFERR select modules linked by the MVS OS/VS linkage editor with invalid APF length indicators.

NOAPFERR select modules linked by the MVS OS/VS linkage editor with valid APF length indicators or modules linked with previous linkage editors.

APPARENTALIAS select main members whose start addresses match the start address of another main member.

NOAPPARENTALIAS select alias members or main members whose start addresses do not match the start address of another main member.

AUTH select APF authorized modules.

NOAUTH select non-authorized modules.

BELOW(Count2) select members with less than the number of lines (amount of storage for load modules) defined by the number, Count2.

BLOCKERR select members with blocksizes exceeding the DCB BLKSIZE.

MAXBLK(sizeb) select members with blocksizes exceeding sizeb.

NOBLOCKERR select members without blocksize errors.

DC select downward-compatible modules.

NODC select no downward-compatible modules.

EDIT select modules which can be reprocessed by the linkage editor.

NOEDIT select modules which can not be reprocessed by the linkage editor.

EXEC select executable modules.

NOEXEC select non-executable modules.

EXTERN select load modules with missing external symbols. The MODULE keyword may also be specified to search for specific missing symbols.

WKEXTERN select load modules missing weak external symbols. The MODULE keyword may also be specified to search for specific missing symbols.

PSEUDOREG select load modules with PSEUDO register symbols. The MODULE keyword may also be specified to search for specific symbols.

LABELREF select load modules with ENTRY symbols. The MODULE keyword may also be specified to search for specific symbols.

COMMON select load modules with COMMON area symbols. The MODULE keyword may also be specified to search for specific symbols.

PRIVATE select load modules with PRIVATE area symbols. The MODULE keyword may also be specified to search for specific symbols.

CSECT select load modules with CSECT symbols. The MODULE keyword may also be specified to search for specific symbols.

FLEVEL select modules processed by the F-level linkage editor.

NOFLEVEL select modules not processed by the F-level linkage editor.

HASALIAS select main members which have one or more aliases.

NOHASALIAS select alias members or main members without any aliases.

ID(Puid) select members with ISPF statistics and userids matching the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. ID is an alias of USERID.

NOID select members without ISPF statistics. NOID is an alias of NOUSERID.

NOTID(Puid) select members without ISPF statistics or members with ISPF statistics and userids which do not match the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. NOTID is an alias of NOTUSERID.

IDRFULL select members which can not contain more IMASPZAP IDR records.

NOIDRFULL select members which can contain more IMASPZAP IDR records.

IOERR select members with I/O errors.

NOIOERR select members without I/O errors.

LKED(Partl) select modules identified as linked by the linkage editor defined by the partial name, **Partl**. **Partl** may be entered as a partial linkage editor IDR name with one to ten characters.

LKEDERR select modules with any of the following illegal linkage edit attribute combinations:

1. Modules with RENT and NOREUS attributes
2. Modules with RMODEANY and AMODE24 or AMODEANY attributes
3. Modules with OVLY and RENT, REUS, REFR, SCTR, RMODEANY, AMODE31, or AMODEANY attributes
4. Modules with TEST and NOEDIT attributes
5. Modules with REUS and SCTR attributes

NOLKEDERR select modules with none of the above illegal linkage edit attribute combinations.

LOADERR select modules which ABEND when they are LOADED.

NOLOADERR select modules which can be LOADED without errors.

LOADONLY select modules marked for LOAD ONLY.

NOLOADONLY select modules not marked for LOAD ONLY.

LRECLERR select members with any of the following LRECL errors:

1. BLKSIZE divided by input LRECL not integral (RECFM=F)
2. Input LRECL exceeding maximum DCB LRECL (RECFM=V)
3. Input LRECL less than 4 bytes (RECFM=V)

NOLRECLERR select members with none of the above LRECL errors.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT or ENTRY names for attribute searches. Note that if this parameter is entered with any TRANS, SYSMOD/USERDATA or ZAP keywords, only those CSECTs which satisfy the MODULE keyword are checked for the other keyword conditions.

The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.

MODULE(Fullm) consider only modules containing a CSECT or ENTRY named FULLM.

MODULE(Partm*) consider only modules containing a CSECT or ENTRY named PARTM...

NAMEERR select members whose member names are not upper case alphameric or whose first character is numeric.

NONAMEERR select members whose member names are upper case alphameric and whose first character is not numeric.

NULL select members with no data.

NONULL select members with some data.

ORPHAN select alias members which have no associated main member.

NOORPHAN select main members or alias members which have an associated main member.

OVERLAY select overlay modules.

NOOVERLAY select non-overlay modules.

PACKED specifies that ISPF packed source members are desired.

NOPACKED specifies that ISPF packed source members are not desired.

PAGE select modules marked for loading on a page boundary.

NOPAGE select modules not marked for loading on a page boundary.

REFR select refreshable modules.

NOREFR select non-refreshable modules.

RENT select reentrant modules.

NORENT select non-reentrant modules.

REUS select reusable modules.

NOREUS select non-reusable modules.

RLDERR select modules linked since OS/VS whose RLD/CONTROL count from the directory does not match the first RLD entry.

NORLDERR select modules whose RLD/CONTROL count from the directory match the first RLD entry or modules linked before OS/VS.

RLDZERO select modules whose RLD/CONTROL count is zero.

NORLDZERO select modules whose RLD/CONTROL count is not zero.

RMODE24 select modules with residence mode 24.

NORMODE24 select modules with residence mode ANY.

RMODEANY select modules with residence mode ANY.

NORMODEANY select modules with residence mode 24.

SCTR select scatter-loaded modules.

NOSCTR select no scatter-loaded modules.

SPFEDIT select members which are currently in-use by an ISPF EDIT session.

NOSPFEDIT select members which are not currently in-use by an ISPF EDIT session.

SSI(hexdata) select members with matching SSI data. This is implemented as a generic match of SSI characters on the first part of the SSI field for the number of characters entered. For example, SSI(698) would match actual SSI fields of 698BADEF or 69800000 but it would not match 06980000.

SSI select members with SSI data.

NOSSI select members without SSI data.

PARTSSI(hx) select members with matching SSI data. This is implemented as a pattern check, matching SSI digits as a string anywhere in the SSI field of the member. For example, PARTSSI(698) would match actual SSI fields of 698BADEF or 00006980 but it would not match 80000069.

SYSMOD(Partu) select modules with user IDR data which matches the partial name, **Partu**. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the SYSMOD operand applies only to CSECT's whose names satisfy the MODULE argument.

TEST select modules with the TEST attribute.

NOTEST select modules without the TEST attribute.

TRANS(Partt) select modules with CSECTS identified as having been assembled or compiled by the translator defined by the partial name, **Partt**. **Partt** may be entered as a partial translator IDR name with one to ten characters. Note that if MODULE(...) is also entered, the TRANS operand applies only to CSECT's whose names satisfy the MODULE argument.

TTR(Lttr:Httr) select members whose start address is in the specified TTR range. **Lttr** defaults to 0 and may be entered as a hexadecimal TTR value from 0 through FFFFFF. **Httr** defaults to FFFFFF and may be entered as a hexadecimal TTR value from 0 through FFFFFF.

USERDATA(Partu) select modules with user IDR data which matches the partial name, **Partu**. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the USERDATA operand applies only to CSECT's whose names satisfy the MODULE argument.

USERID(Puid) select members with ISPF statistics and userids matching the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. USERID is an alias of ID.

NOUSERID select members without ISPF statistics.   NOUSERID is an alias of NOID.

NOTUSERID(Puid) select members without ISPF statistics or members with ISPF statistics and userids which do not match the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. NOTUSERID is an alias of NOTID.

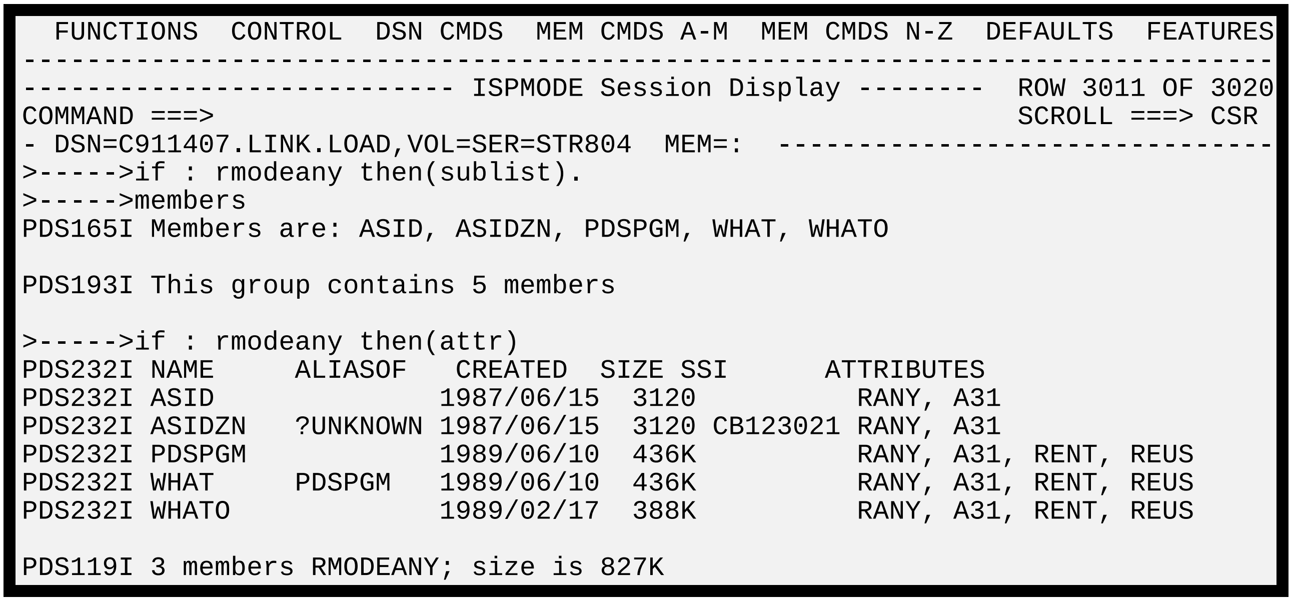
VSLKED select modules linked by the MVS OS/VS linkage editor.

NOVSLKED select modules not linked by the MVS OS/VS linkage editor.

ZAP(Partz) select modules with zap IDR data which matches the partial name, **Partz**. **Partz** may be entered as partial zap IDR data with one to eight characters. Note that if MODULE(...) is also entered, the ZAP operand applies only to CSECT's whose names satisfy the MODULE argument.

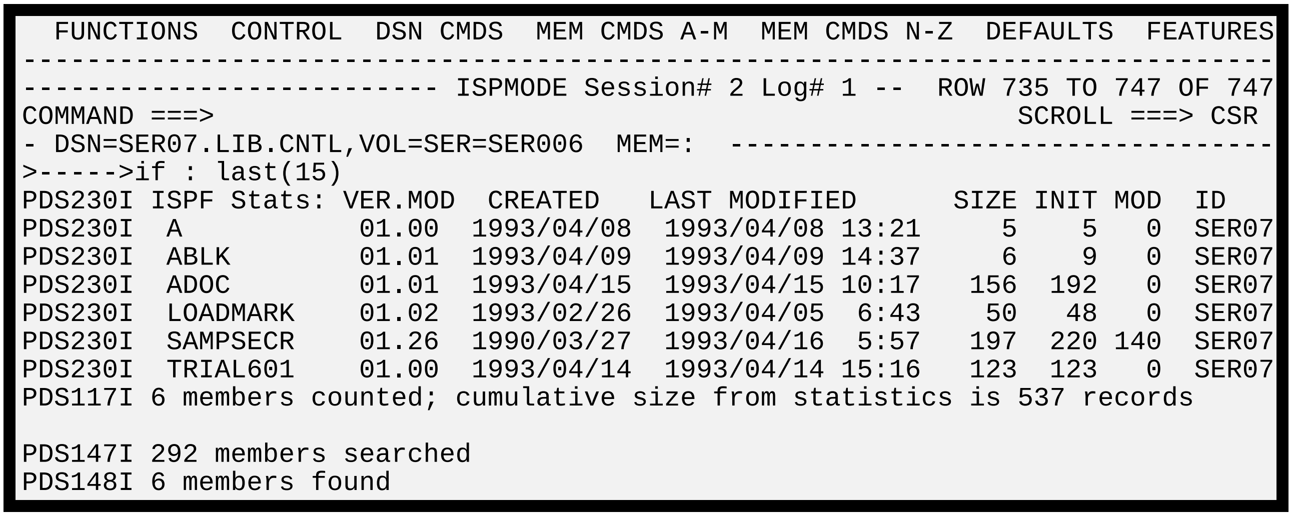
THEN(action) execute a subcommand if all conditions are met for a member. The subcommand may be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FIND, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY or VIEW. Note that NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET. If neither a THEN nor an ELSE operand is entered, THEN(ATTRIB) is the default condition.

ELSE(action) execute a subcommand if any condition is <u>not</u> met for a member. The subcommand may be ATTRIB, BROWSE, DIRENTRY, DELETE, EDIT, EXCLUDE, FIND, FSE, HISTORY, LIST, MAP, MEMBERS, MEMLIST, NEWML, PRINT, REVIEW, SUBMIT, SUBLIST, TSOEDIT, TSOLIST, VERIFY or VIEW. Note that NEWML is similar to MEMLIST except that it also performs a MEMLIST RESET. If neither a THEN nor an ELSE operand is entered, THEN(ATTRIB) is the default condition.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display --------  ROW 3011 OF 3020
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=:  -------------------------------
>----->if : rmodeany then(sublist).
>----->members
PDS165I Members are: ASID, ASIDZN, PDSPGM, WHAT, WHATO

PDS193I This group contains 5 members

>----->if : rmodeany then(attr)
PDS232I NAME     ALIASOF   CREATED  SIZE SSI      ATTRIBUTES
PDS232I ASID              1987/06/15  3120          RANY, A31
PDS232I ASIDZN   ?UNKNOWN 1987/06/15  3120 CB123021 RANY, A31
PDS232I PDSPGM            1989/06/10  436K          RANY, A31, RENT, REUS
PDS232I WHAT     PDSPGM   1989/06/10  436K          RANY, A31, RENT, REUS
PDS232I WHATO             1989/02/17  388K          RANY, A31, RENT, REUS

PDS119I 3 members RMODEANY; size is 827K
```

Figure 58. Sample IF Subcommand (load)



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 2 Log# 1 --  ROW 735 TO 747 OF 747
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=:  ----------------------------------
>----->if : last(15)
PDS230I ISPF Stats: VER.MOD  CREATED   LAST MODIFIED      SIZE INIT MOD  ID
PDS230I  A           01.00  1993/04/08  1993/04/08 13:21     5    5   0  SER07
PDS230I  ABLK        01.01  1993/04/09  1993/04/09 14:37     6    9   0  SER07
PDS230I  ADOC        01.01  1993/04/15  1993/04/15 10:17   156  192   0  SER07
PDS230I  LOADMARK    01.02  1993/02/26  1993/04/05  6:43    50   48   0  SER07
PDS230I  SAMPSECR    01.26  1990/03/27  1993/04/16  5:57   197  220 140  SER07
PDS230I  TRIAL601    01.00  1993/04/14  1993/04/14 15:16   123  123   0  SER07
PDS117I 6 members counted; cumulative size from statistics is 537 records

PDS147I 292 members searched
PDS148I 6 members found
```

Figure 59. Sample IF Subcommand (source)

ISPF Subcommand

Purpose The ISPF subcommand invokes the ISPF primary panel with any option.

Example ISPF   3.3

Syntax

```
ISPF   [option]
```

Aliases IS, ISP, ISPF

Defaults none

Required none

Operands

option specifies an initial ISPF option (such as **3.3**) which allows intermediate panel displays to be skipped.

Remarks Note that the result of using the ISPF subcommand is a temporary suspension of your STARWARP session and entry to the specified ISPF function. After you eventually terminate ISPF by repeated ENDs, your STARWARP session will be activated again. This is a stacked ISPF session.

ISPMODE Function

Purpose The ISPMODE subcommand switches STARWARP to a full screen session display mode. Subcommands entered and the outputs from internal STARWARP subcommands are maintained in a log.

Example ISPMODE

Syntax

```
ISPMODE
```

Aliases ISPM, ISPMO, ISPMOD, ISPMODE

Operands (no operands are supported for the ISPMODE subcommand).

Remarks The ISPMODE log is displayed in response to a ISPMODE subcommand issued in line mode; you then enter ISPMODE (an ISPF dialog mode). After you are in dialog mode, you can use the LOG command to select any one of nine different logs for your session display.

When ISPMODE is active, STARWARP operates as a normal ISPF dialog. You may enter the **HELP** command while in ISPMODE to get ISPF tutorial information or enter the **INDEX** command to go to the STARWARP tutorial index panel directly.

You may suspend ISPMODE with the **SUSPEND** command. SUSPEND will place your session into line mode (for example, you can go into line mode to SEND or RECEIVE data using IND$FILE). When you want to enter ISPMODE again, enter an **ISPMODE**, **ISPXEQ** or **MEMLIST** subcommand.

For additional information on the log and ISPMODE processing, please refer to **LOG Function** on page 158.

ISPXEQ Function

Purpose The ISPXEQ subcommand is normally used to enter STARWARP in LISTA or LISTV dialog mode instead of the usual ISPMODE or MEMLIST mode.

ISPXEQS is a variation of ISPXEQ which may be used to invoke a "short" variation and STARWARP will terminate at the first END command.

Example STARWARP LIB.CNTL ISPXEQ LISTA OPEN

Syntax

```
ISPXEQ / ISPXEQS   {command}
```

Aliases ISPX, ISPXE, ISPXEQ, ISPXEQS

Defaults none

Required command

Operands

command specifies a ISPMODE command to be executed after STARWARP initialization.

Remarks The ISPXEQ subcommand is often used in building ISPF command table entries for STARWARP dialog functions. For example, the following command table entry could be added for an ISPF command called LA:
SELECT CMD(STARWARP 'SYS1.HELP' ISPXEQ LISTA &ZPARM)

LIST Subcommand

Purpose The LIST subcommand displays data from a member or data set.

Example LIST   mema:memb

Syntax

```
LIST   memgroup [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP         ]
                [OFFSET(Hx) MODULE({* / Fullm / Partm*)}        ]
                [MAXMEMBERS(num)                                ]
                [FORMAT(from:to,from:to, ...)                   ]
                [SKIPREC(n) MAXIN(n) MAXOUT(n)                  ]
                [SKIPCOL(n) MAXLEN(n)                           ]
              * [FROMKEY(key)/FROMADDRESS(add)/FROMNUMBER(num)  ]
              * [TOKEY(key)/TOADDRESS(add)/TONUMBER(num)        ]
  *NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases LI, LIS, LIST

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults memgroup, NUM or previous LIST/FIND/REPLACE format

Required none

Operands

memgroup identifies the member(s) to be displayed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

NUM examine the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise) of each logical record; if the field is not numeric, switch formatting to NONUM mode for the remainder of the member.

For ISPF-saved members, the high-order six digits of the line number field is formatted; otherwise, the low-order six digits of the line number field is formatted by suppressing leading zeroes. The line number segment is followed by a blank and up to 249 characters of data from a logical record.

SNUM discard the line number field (the last 8 bytes for a fixed-format record or the first 8 otherwise). Display up to 256 bytes from a logical record.

NONUM display up to 256 bytes from a logical record (without regard to line numbers).

LBLOCK format logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record) and segments of up to 64 characters surrounded by asterisks.

Note: for load modules, only CSECT data will be displayed; if a name is entered in a MODULE operand, only those CSECT or ENTRY names which satisfy the condition will be displayed. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset) and segments of up to 64 characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

LDUMP Logical records are formatted with two hexadecimal offsets (displacement in the member and displacement in the logical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks.

Note: for load modules, only CSECT data will be displayed; if a name is entered in a MODULE operand, only those CSECT or ENTRY names which satisfy the condition will be displayed. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset), segments of up to 32 characters of hexadecimal data and up to 16 bytes of characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

BLOCK Physical records are formatted with two hexadecimal offsets (displacement in the member and displacement in the physical record) and segments of up to 64 characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

If control interval processing is being performed, the first field is always the control interval relative byte address.

DUMP Physical records are formatted with two hexadecimal offsets (displacement in the member and displacement in the physical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

If control interval processing is being performed, the first field is always the control interval relative byte address.

OFFSET(Hx) specifies a 1 to 6 digit hexadecimal module offset at which the display is to begin. Note: this operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT or ENTRY names to be displayed. Note: this operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected. The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.

MODULE(Fullm) display only a CSECT or ENTRY named FULLM.

MODULE(Partm*) display only a CSECT or ENTRY named PARTM...

MAXMEMBERS(n) n is coded as 0 through 9999999 to specify the maximum number of members to be output by LIST.

FORMAT(from:to,from:to, ...) specifies the record columns to display in the output when displaying LIST data. FORMAT is only supported with NUM, SNUM or NONUM data. For example, **FORMAT(21:30,0,1:10)** specifies that the output should be formatted with data from columns 21 through 30, a blank and data from columns 1 through 10.

SKIPREC(n) ignore **n** (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) at the beginning of a member.

MAXIN(n) input up to **n** (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) for a member after satisfying any SKIPREC operand.

MAXOUT(n) display up to **n** (coded as 0 through 9999999) output lines for a member.

SKIPCOL(n) ignore **n** (coded as 0 through 99999) columns at the beginning of each logical record (physical record for BLOCK or DUMP formats).

Note: for NUM or SNUM output format with record format V, SKIPCOL(0) refers to the first data position after the line number field.

MAXLEN(n) display up to **n** (coded as 0 through 99999) characters in a logical record (physical record for BLOCK or DUMP formats).

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access begins at the first record whose key matches (or is greater than) the portion of the key specified.

This parameter may be used with TOKEY and it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can be specified for any VSAM data set component if control interval processing is being used.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set. This parameter may be used with

TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified. This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can be specified for any VSAM data set component if control interval processing is being used.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set. This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

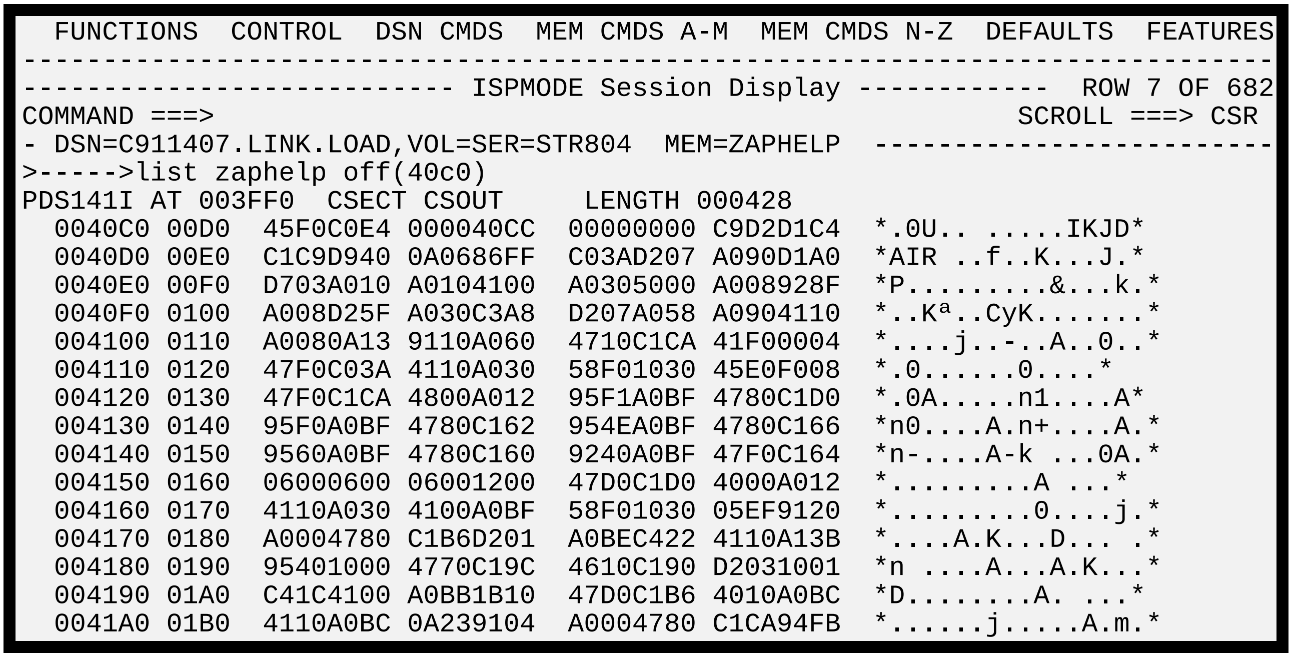
Remarks Unprintable characters for 3270-type devices are translated to periods before they are displayed.

LIST formats are NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK and DUMP. For load modules, the NUM, SNUM and NONUM formats are equivalent to the LDUMP format and for VSAM data sets, the NUM, SNUM and NONUM formats are equivalent to the LBLOCK format.

The default format is initially NUM; however, each time a format operand is entered on a LIST, FIND or REPLACE subcommand, the value entered is used as the output format for subsequent LIST, FIND and REPLACE subcommands. Also, note the following:

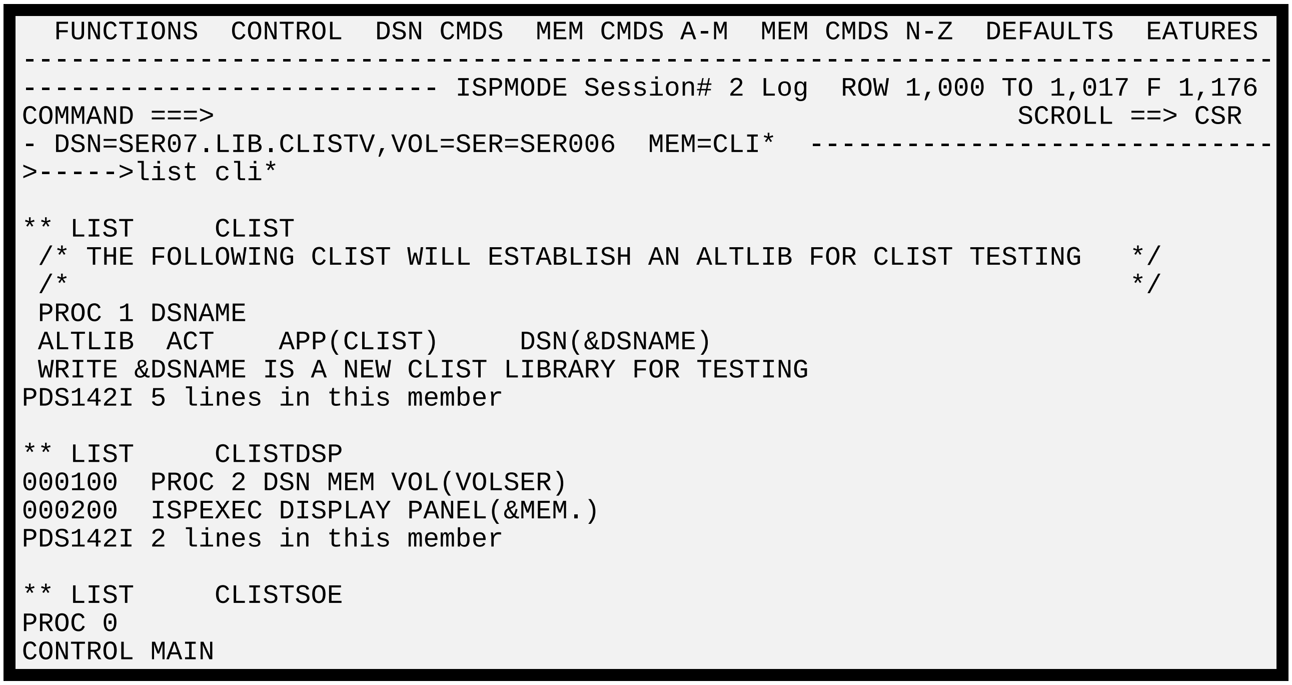
1. Formats NUM, SNUM and NONUM limit the display length for logical records to 256 characters.
2. Formats BLOCK and DUMP apply to physical records for non-VSAM data sets (or when control interval processing is being performed); the other formats apply to logical records. For load modules, LBLOCK and LDUMP formats display only CSECT data.
3. For VSAM DATA or INDEX components, the LIST, FIND and REPLACE subcommands support control interval access using the DUMP or BLOCK display formats. Instead of accessing individual VSAM records, each GET or PUT obtains a VSAM control interval.

   Control interval access could be useful if a VSAM data set has logical errors. REPLACE could be used to repair the error; however, since only the component is opened for update, the next access of the data set through the related cluster will get warning errors due to the differing time stamps.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ------------  ROW 7 OF 682
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAPHELP  -------------------------
>----->list zaphelp off(40c0)
PDS141I AT 003FF0  CSECT CSOUT     LENGTH 000428
  0040C0 00D0  45F0C0E4 000040CC  00000000 C9D2D1C4  *.0U.. .....IKJD*
  0040D0 00E0  C1C9D940 0A0686FF  C03AD207 A090D1A0  *AIR ..f..K...J.*
  0040E0 00F0  D703A010 A0104100  A0305000 A008928F  *P.........&...k.*
  0040F0 0100  A008D25F A030C3A8  D207A058 A0904110  *..Kª..CyK.......*
  004100 0110  A0080A13 9110A060  4710C1CA 41F00004  *....j..-..A..0..*
  004110 0120  47F0C03A 4110A030  58F01030 45E0F008  *.0......0....*
  004120 0130  47F0C1CA 4800A012  95F1A0BF 4780C1D0  *.0A.....n1....A*
  004130 0140  95F0A0BF 4780C162  954EA0BF 4780C166  *n0....A.n+....A.*
  004140 0150  9560A0BF 4780C160  9240A0BF 47F0C164  *n-....A-k ...0A.*
  004150 0160  06000600 06001200  47D0C1D0 4000A012  *.........A ...*
  004160 0170  4110A030 4100A0BF  58F01030 05EF9120  *.........0....j.*
  004170 0180  A0004780 C1B6D201  A0BEC422 4110A13B  *....A.K...D... .*
  004180 0190  95401000 4770C19C  4610C190 D2031001  *n ....A...A.K...*
  004190 01A0  C41C4100 A0BB1B10  47D0C1B6 4010A0BC  *D........A. ...*
  0041A0 01B0  4110A0BC 0A239104  A0004780 C1CA94FB  *......j.....A.m.*
```

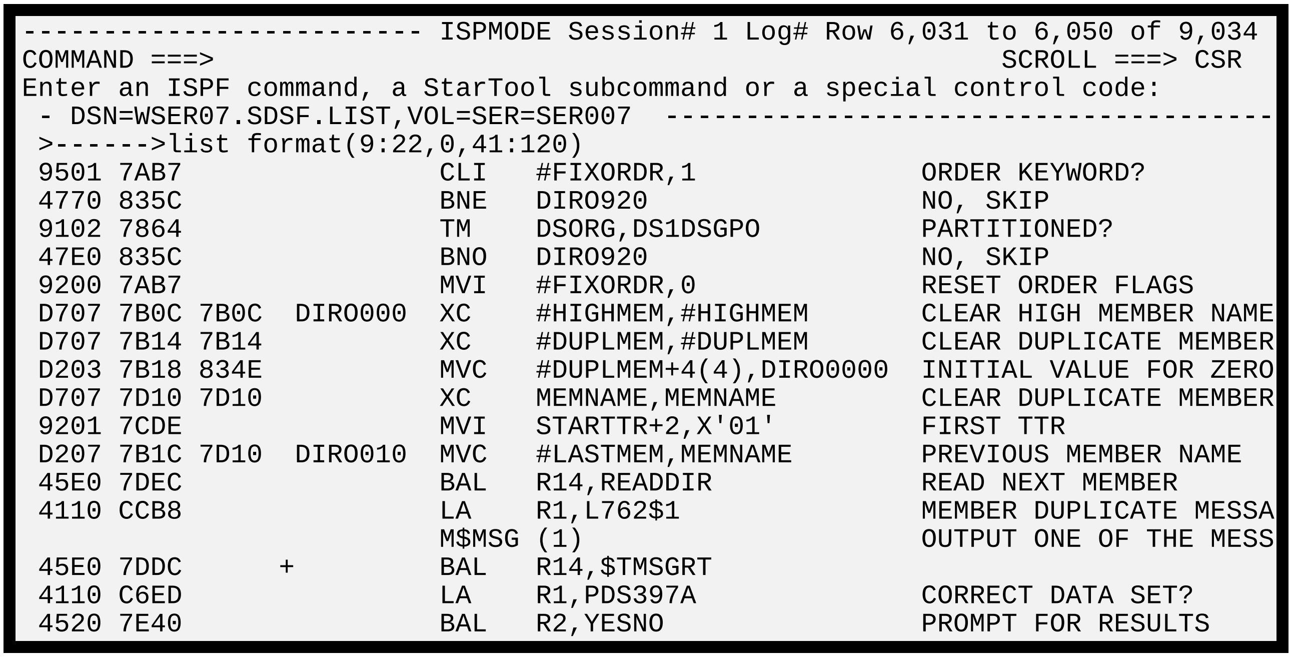
Figure 60. Sample LIST Subcommand (load with LDUMP)



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  EATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 2 Log  ROW 1,000 TO 1,017 F 1,176
COMMAND ===>                                                  SCROLL ==> CSR
- DSN=SER07.LIB.CLISTV,VOL=SER=SER006  MEM=CLI*  -----------------------------
>----->list cli*

** LIST     CLIST
 /* THE FOLLOWING CLIST WILL ESTABLISH AN ALTLIB FOR CLIST TESTING   */
 /*                                                                  */
 PROC 1 DSNAME
 ALTLIB  ACT    APP(CLIST)     DSN(&DSNAME)
 WRITE &DSNAME IS A NEW CLIST LIBRARY FOR TESTING
PDS142I 5 lines in this member

** LIST     CLISTDSP
000100  PROC 2 DSN MEM VOL(VOLSER)
000200  ISPEXEC DISPLAY PANEL(&MEM.)
PDS142I 2 lines in this member

** LIST     CLISTSOE
PROC 0
CONTROL MAIN
```

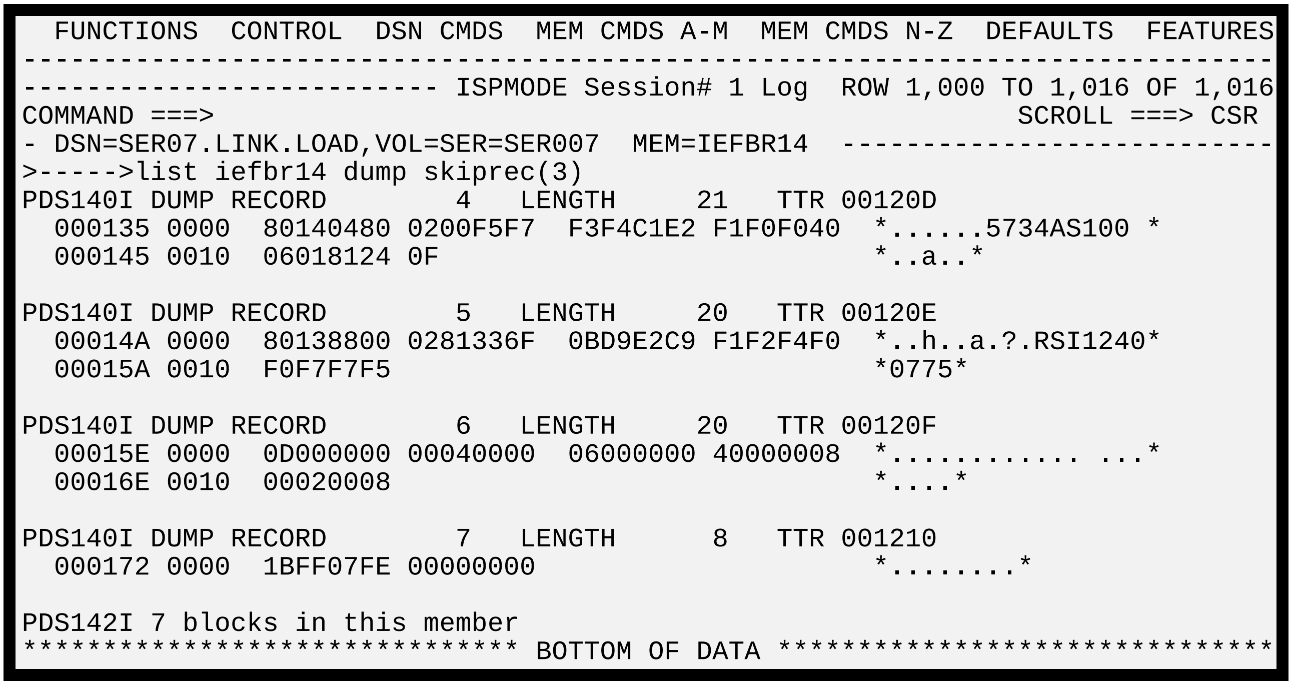
Figure 61. Sample LIST Subcommand (source)

LIST Subcommand



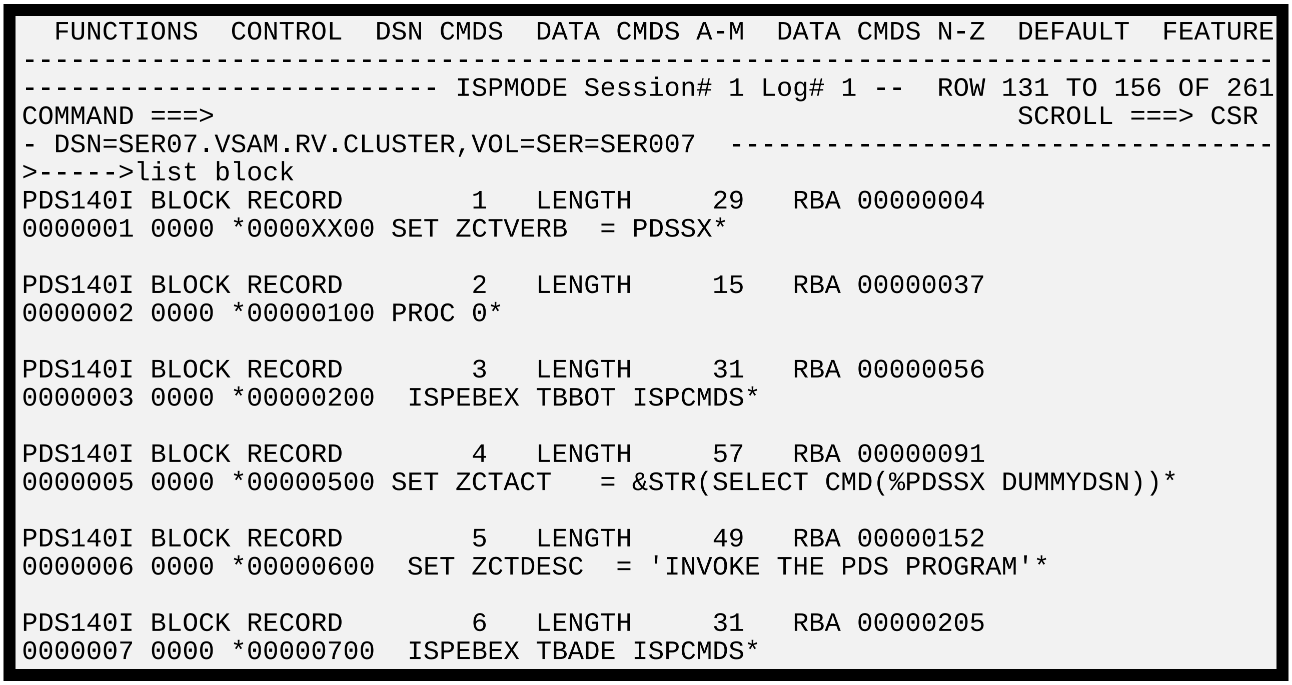
```
------------------------- ISPMODE Session# 1 Log# Row 6,031 to 6,050 of 9,034
COMMAND ===>                                                     SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
 - DSN=WSER07.SDSF.LIST,VOL=SER=SER007  ------------------------------------
 >------>list format(9:22,0,41:120)
 9501 7AB7                CLI   #FIXORDR,1              ORDER KEYWORD?
 4770 835C                BNE   DIRO920                 NO, SKIP
 9102 7864                TM    DSORG,DS1DSGPO          PARTITIONED?
 47E0 835C                BNO   DIRO920                 NO, SKIP
 9200 7AB7                MVI   #FIXORDR,0              RESET ORDER FLAGS
 D707 7B0C 7B0C  DIRO000  XC    #HIGHMEM,#HIGHMEM       CLEAR HIGH MEMBER NAME
 D707 7B14 7B14           XC    #DUPLMEM,#DUPLMEM       CLEAR DUPLICATE MEMBER
 D203 7B18 834E           MVC   #DUPLMEM+4(4),DIRO0000  INITIAL VALUE FOR ZERO
 D707 7D10 7D10           XC    MEMNAME,MEMNAME         CLEAR DUPLICATE MEMBER
 9201 7CDE                MVI   STARTTR+2,X'01'         FIRST TTR
 D207 7B1C 7D10  DIRO010  MVC   #LASTMEM,MEMNAME        PREVIOUS MEMBER NAME
 45E0 7DEC                BAL   R14,READDIR             READ NEXT MEMBER
 4110 CCB8                LA    R1,L762$1               MEMBER DUPLICATE MESSA
                          M$MSG (1)                     OUTPUT ONE OF THE MESS
 45E0 7DDC      +         BAL   R14,$TMSGRT
 4110 C6ED                LA    R1,PDS397A              CORRECT DATA SET?
 4520 7E40                BAL   R2,YESNO                PROMPT FOR RESULTS
```

Figure 62. Sample LIST Subcommand (sequential with FORMAT)



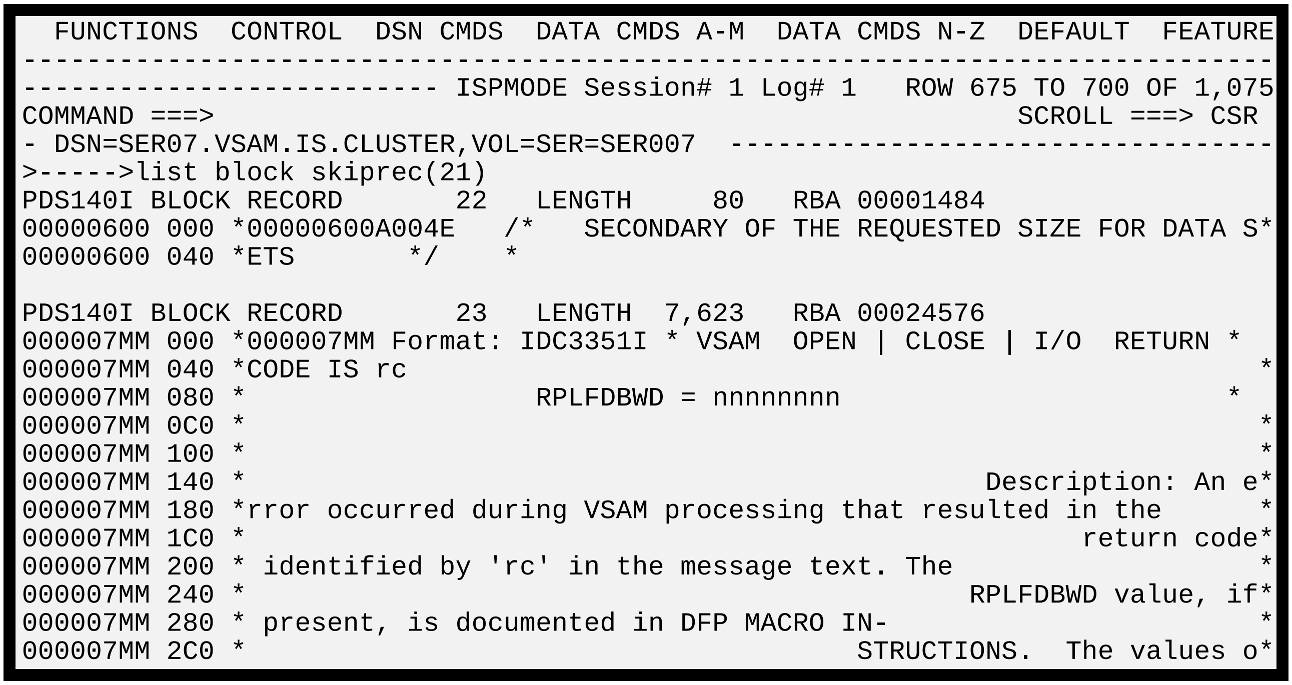
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,016 OF 1,016
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=IEFBR14  --------------------------
>----->list iefbr14 dump skiprec(3)
PDS140I DUMP RECORD        4   LENGTH     21   TTR 00120D
  000135 0000  80140480 0200F5F7  F3F4C1E2 F1F0F040  *......5734AS100 *
  000145 0010  06018124 0F                           *..a..*

PDS140I DUMP RECORD        5   LENGTH     20   TTR 00120E
  00014A 0000  80138800 0281336F  0BD9E2C9 F1F2F4F0  *..h..a.?.RSI1240*
  00015A 0010  F0F7F7F5                              *0775*

PDS140I DUMP RECORD        6   LENGTH     20   TTR 00120F
  00015E 0000  0D000000 00040000  06000000 40000008  *............ ...*
  00016E 0010  00020008                              *....*

PDS140I DUMP RECORD        7   LENGTH      8   TTR 001210
  000172 0000  1BFF07FE 00000000                     *........*

PDS142I 7 blocks in this member
****************************** BOTTOM OF DATA ******************************
```

Figure 63. Sample LIST Subcommand (load with DUMP)



```
  FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1 --  ROW 131 TO 156 OF 261
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.VSAM.RV.CLUSTER,VOL=SER=SER007  ----------------------------------
>----->list block
PDS140I BLOCK RECORD        1   LENGTH     29   RBA 00000004
0000001 0000 *0000XX00 SET ZCTVERB  = PDSSX*

PDS140I BLOCK RECORD        2   LENGTH     15   RBA 00000037
0000002 0000 *00000100 PROC 0*

PDS140I BLOCK RECORD        3   LENGTH     31   RBA 00000056
0000003 0000 *00000200  ISPEBEX TBBOT ISPCMDS*

PDS140I BLOCK RECORD        4   LENGTH     57   RBA 00000091
0000005 0000 *00000500 SET ZCTACT   = &STR(SELECT CMD(%PDSSX DUMMYDSN))*

PDS140I BLOCK RECORD        5   LENGTH     49   RBA 00000152
0000006 0000 *00000600  SET ZCTDESC  = 'INVOKE THE PDS PROGRAM'*

PDS140I BLOCK RECORD        6   LENGTH     31   RBA 00000205
0000007 0000 *00000700  ISPEBEX TBADE ISPCMDS*
```

Figure 64. Sample LIST Subcommand (Variable RRDS)



```
  FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1   ROW 675 TO 700 OF 1,075
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.VSAM.IS.CLUSTER,VOL=SER=SER007  ----------------------------------
>----->list block skiprec(21)
PDS140I BLOCK RECORD       22   LENGTH     80   RBA 00001484
00000600 000 *00000600A004E   /*   SECONDARY OF THE REQUESTED SIZE FOR DATA S*
00000600 040 *ETS       */    *

PDS140I BLOCK RECORD       23   LENGTH  7,623   RBA 00024576
000007MM 000 *000007MM Format: IDC3351I * VSAM  OPEN | CLOSE | I/O  RETURN *
000007MM 040 *CODE IS rc                                                     *
000007MM 080 *                  RPLFDBWD = nnnnnnnn                        *
000007MM 0C0 *                                                               *
000007MM 100 *                                                               *
000007MM 140 *                                              Description: An e*
000007MM 180 *rror occurred during VSAM processing that resulted in the      *
000007MM 1C0 *                                                    return code*
000007MM 200 * identified by 'rc' in the message text. The                   *
000007MM 240 *                                             RPLFDBWD value, if*
000007MM 280 * present, is documented in DFP MACRO IN-                       *
000007MM 2C0 *                                      STRUCTIONS.  The values o*
```

Figure 65. Sample LIST Subcommand (VSAM KSDS)

LISTA Function

Purpose The LA or LISTA command is used to display your TSO session allocations in an ISPF table. DDNAME is another way to use the LISTA table. The LISTA/DDNAME table is maintained in DDNAME and concatenation order (the SORT command is not supported).

Example LISTA open old mod

Syntax

```
LISTA   [ALL              ]
        [OPEN             ]
        [UNOPEN/CLOSED    ]
        [CONCAT/CATENATE  ]
        [NONCAT/NONCON    ]
        [DUMMY/NULLFILE   ]
        [ATTRIB           ]
        [TERMINAL         ]
        [JESFILE/SYSOUT   ]
        [TMP/TEMPORY      ]
        [NEW MOD SHR OLD  ]
        [TASKLIB          ]
        [SETDEF           ]
        [RESET            ]
```

Aliases LA, LISTA

Defaults ALL

Required none

Operands

ALL select all allocations (initial default).
OPEN select open allocations.
UNOPEN select closed allocations.
CLOSED select closed allocations.
CONCAT select allocations which are a part of a concatenation.
CATENATE select allocations which are a part of a concatenation.
NONCAT select allocations which are not concatenated.
NONCON select allocations which are not concatenated.
DUMMY select NULLFILE allocations.
NULLFILE select NULLFILE allocations.
ATTRIB select NULLFILE allocations which are not also TERMINAL.
TERMINAL select TERMINAL allocations.
JESFILE select allocations to JES output files.
SYSOUT select allocations to JES output files.
TMP select temporary allocations.
TEMPORY select temporary allocations.
NEW select NEW allocations.
MOD select MOD allocations.
SHR select SHR allocations.
OLD select OLD allocations.
TASKLIB select TASKLIB allocations (including STEPLIB).
SETDEF make the selected operands the new default for this session instead of ALL.

RESET clear the LISTA table before adding any new entries.

Remarks

The LISTA/DDNAME ISPF table is displayed in response to a LISTA command. When you are in a LISTA/DDNAME display, you have many options: you may delete a part of the table, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

LISTA has alternate panels, as do most of the ISPMODE tables in STARWARP, containing extra information about the listed data sets. The first LISTA/DDNAME panel shows basic allocation information such as DDNAME, data set TYPE, Open count (under the header O#), VOLUME allocated, MEMBER name and DSNAME. Command error messages are provided under the DATA/MSG header as follows:

***INVALID**	this line command is not supported.
***DD GONE**	this file is not allocated.
***OPEN DD**	this file is open.
***IN USE**	this file is in-use.
***NOT CON**	this file is not concatenated.
***PERMCON**	this file is permanently concatenated.
NO DDN	this file has a blank ddname.

TYPE will be the data set DSORG or "VIO", "CTLG", "VVDS", "JES" or "TERM" depending on the type of allocation.

The second LISTA/DDNAME panel is a double line panel which includes some additional information: DISP for data set disposition and STAT for data set status. The STAT field will be **TMP** for temporary, **PRM** for permanent or **CNV** for convertible.

The following primary commands are supported directly for the LISTA/DDNAME function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

ALT[ERNAT] displays an alternate view of the LISTA/DDNAME table.

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

BAT[CHJCL] builds batch JCL which will include all DD statements in the LISTA table.

DUA[L] displays a double line view of the LISTA/DDNAME table.

DD[NAME] rebuilds the LISTA/DDNAME table selecting data sets with a DDNAME mask.
Syntax: `DDNAME ddnamemask`

EDITT[BL] (or **ET[BL]**) enters an edit session on LISTA/DDNAME table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
  [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
  [ANY/DDNAME/DSNAME/VOLUME/MEMBER/ALLOC]
```

FI[ND] global command, changes to each data set in the table and issues a FIND subcommand. Since the syntax entered on a FIND subcommand is actually applied to each data set individually, you should not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.
Syntax:
```
FIND memgroup 'anystring'
     [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
     [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
```

GL[OBAL] global command, changes to each data set in the table and issues a STARWARP subcommand.
Syntax: `GLOBAL anysubcommand [anyoperands]`

LE[FT] rotates through alternate views of the LISTA/DDNAME table. PF keys 10 and 22 are normally set to LEFT.

LISTA rebuilds the LISTA/DDNAME table selecting data sets by attributes.
Syntax: `LISTA [attributes]`

MODEL global command, changes to each data set in the table and issues a MODEL command.

NOR[MAL] displays the default view of the LISTA/DDNAME table.

OUT[PUT] outputs the LISTA/DDNAME table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the LISTA/DDNAME table based on a string match.
Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
          [ANY/DDNAME/DSNAME/VOLUME/MEMBER/ALLOC]
```

REPL[ACE] global command, changes to each data set in the table and issues a REPLACE subcommand. Since the syntax entered on a REPLACE subcommand is actually applied to each data set individually, you should not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.
Syntax:
```
REPLACE memgroup 'fromstring' 'tostring'
          [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
          [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
          [WRITE/NOWRITE]
```

RESET often used with data set tagging, clears the DATA/MSG field in all table entries.

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the LISTA/DDNAME table. PF keys 11 and 23 are normally set to RIGHT.

SEEK global command, changes to each data set in the table and looks for a member.
Syntax: `SEEK member`

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

X clears the LISTA/DDNAME table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the LISTA/DDNAME table; this is equivalent to **X ALL**

The following line commands are supported in the LISTA/DDNAME function:

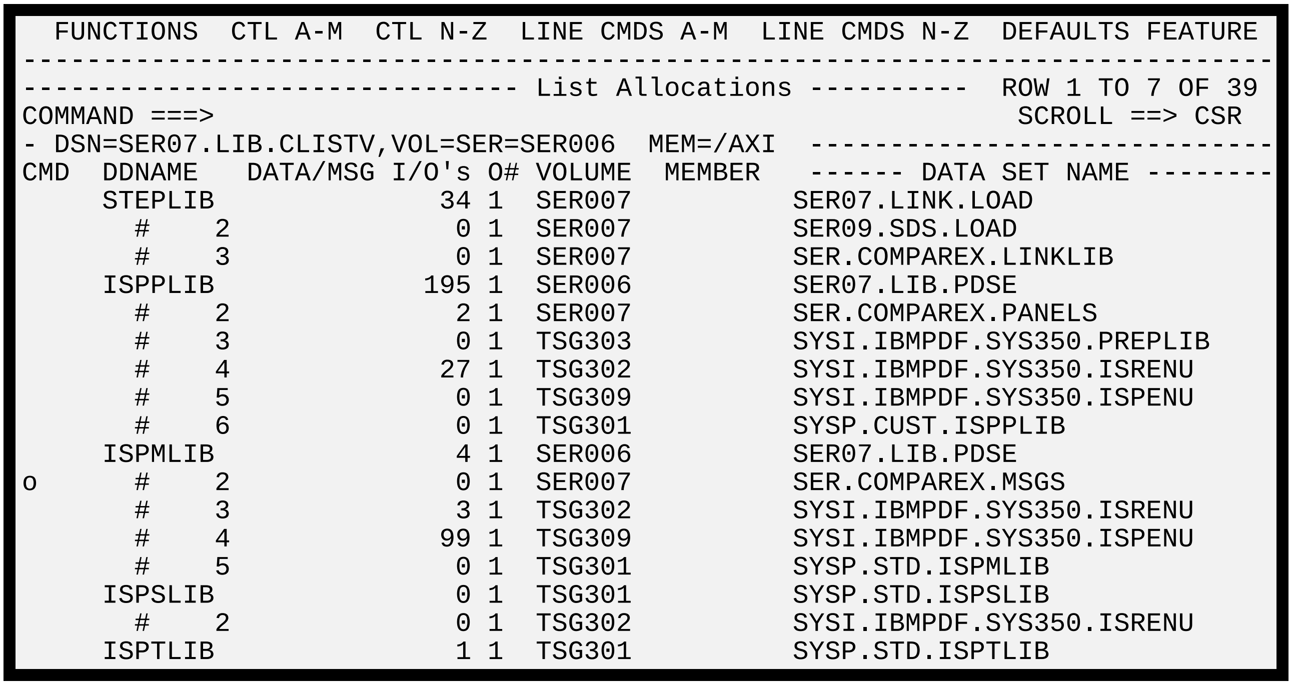
+ provide an extension panel for command entry.
= repeat the previous line command.
6 change to the data set and MEMLIST all.
A change to the data set and execute an alternate subcommand.
B change to the data set and MEMLIST all.
C change to the data set.
DCAT deconcatenate non-permanent data sets in a concatenated group.
E change to the data set and MEMLIST all.
FREE free a DDNAME or all data sets in a concatenated group.
GO change to the data set using GO processing (a number is optional).
IDC perform an IDCAMS LISTC and direct output to the log.
K kill and clear all following line commands.
LC add the data set name to the current LISTC/LISTF table.
LOG copy the line into the log.
M provide line command selection and entry assistance.

ML change to the data set and MEMLIST all.
MOD change to the data set and MODEL.
O provide line command selection and operand syntax assistance.
S select line command (normally CHANGE, it is set by SETSEL).
SEEK change to the data set and check for a member.
SETA save a STARWARP subcommand for repeated use by the ALT line command.
TAG mark this table entry with *TAG* in the DATA/MSG field.
U change to the data set and display USAGE.
UT select the extended user line command panel.
W add the data set name (and any member name) to the WORKPAD.
WHO check for users of this data set with the WHOHAS subcommand.
X drop the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTA/DDNAME function:

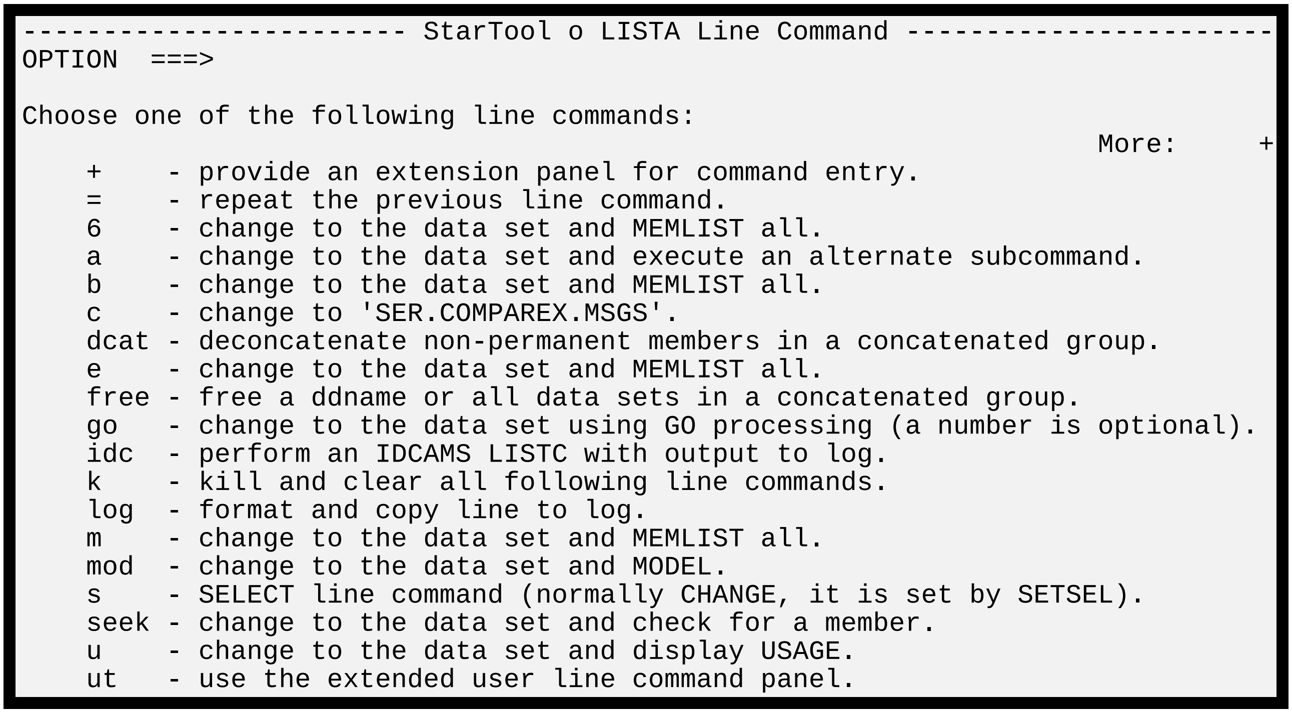
== = command, repeat the previous line command for the range of lines.
AA A command, change and execute an alternate subcommand for each line.
BB B command, change and MEMLIST all for each line.
CC C command, change data set for each line.
EE E command, change and MEMLIST all for each line.
GG GO command, change to each data set using GO processing.
LL LOG command, copy the range of lines into the log.
MM M command, change and MEMLIST all for each line.
OO O command, provide line command assistance for each line.
SS S command, select each line in the range of table lines.
SSEE SEEK command, change to each data set and check for a member.
UU U command, change and display USAGE for each line.
XX X command, drop the range of table lines.

LISTA Function



```
  FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS FEATURE
------------------------------------------------------------------------------
------------------------------ List Allocations ----------  ROW 1 TO 7 OF 39
COMMAND ===>                                                 SCROLL ==> CSR
- DSN=SER07.LIB.CLISTV,VOL=SER=SER006  MEM=/AXI  ----------------------------
CMD  DDNAME   DATA/MSG I/O's O# VOLUME  MEMBER   ------ DATA SET NAME --------
     STEPLIB              34 1  SER007           SER07.LINK.LOAD
       #    2              0 1  SER007           SER09.SDS.LOAD
       #    3              0 1  SER007           SER.COMPAREX.LINKLIB
     ISPPLIB             195 1  SER006           SER07.LIB.PDSE
       #    2              2 1  SER007           SER.COMPAREX.PANELS
       #    3              0 1  TSG303           SYSI.IBMPDF.SYS350.PREPLIB
       #    4             27 1  TSG302           SYSI.IBMPDF.SYS350.ISRENU
       #    5              0 1  TSG309           SYSI.IBMPDF.SYS350.ISPENU
       #    6              0 1  TSG301           SYSP.CUST.ISPPLIB
     ISPMLIB               4 1  SER006           SER07.LIB.PDSE
o      #    2              0 1  SER007           SER.COMPAREX.MSGS
       #    3              3 1  TSG302           SYSI.IBMPDF.SYS350.ISRENU
       #    4             99 1  TSG309           SYSI.IBMPDF.SYS350.ISPENU
       #    5              0 1  TSG301           SYSP.STD.ISPMLIB
     ISPSLIB               0 1  TSG301           SYSP.STD.ISPSLIB
       #    2              0 1  TSG302           SYSI.IBMPDF.SYS350.ISRENU
     ISPTLIB               1 1  TSG301           SYSP.STD.ISPTLIB
```

Figure 66. Sample LISTA Table



```
----------------------- StarTool o LISTA Line Command -----------------------
OPTION  ===>

Choose one of the following line commands:
                                                                 More:     +
    +    - provide an extension panel for command entry.
    =    - repeat the previous line command.
    6    - change to the data set and MEMLIST all.
    a    - change to the data set and execute an alternate subcommand.
    b    - change to the data set and MEMLIST all.
    c    - change to 'SER.COMPAREX.MSGS'.
    dcat - deconcatenate non-permanent members in a concatenated group.
    e    - change to the data set and MEMLIST all.
    free - free a ddname or all data sets in a concatenated group.
    go   - change to the data set using GO processing (a number is optional).
    idc  - perform an IDCAMS LISTC with output to log.
    k    - kill and clear all following line commands.
    log  - format and copy line to log.
    m    - change to the data set and MEMLIST all.
    mod  - change to the data set and MODEL.
    s    - SELECT line command (normally CHANGE, it is set by SETSEL).
    seek - change to the data set and check for a member.
    u    - change to the data set and display USAGE.
    ut   - use the extended user line command panel.
```

Figure 67. Sample LISTA line options panel

LISTC Function

Purpose

The LC or LISTC command is used to display data sets from a catalog source in an ISPF table. The table displaying LISTC data sets is combined with LISTF data sets.

LISTC/LISTF tables can be saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE, ID, MERGE, NOSAVE and ERASE commands. To automatically save these tables at program end or when a new LISTC/LISTF table is retrieved, check the setting of "Automatic save" in SETALL for LISTC/LISTF.

Saved LISTC/LISTF tables are given members names of the form **PDSLF0na** if the name is one or two numeric digits; otherwise, a name of the form **##name** is used where name is a one to six character alphameric table identification name.

Note that the **FIND** and **REPLACE** global subcommands bypass non-partitioned data sets. If you want to search or update non-partitioned data sets, you may use **GLOBAL** followed by **FIND** or **REPLACE** and any operands.

In the LISTC prompt panel, enter the LEVEL/DSN field as a partial data set name. The first level may contain an ***** after one or more characters and any additional levels may contain an ***** before or after any characters in that level. For example: **SYS*.NU*.*.*XY**

Due to a restriction in the SUPERLOCATE service used, generic searches for GDG data sets must contain an * in the first position of the GDG data set node. For example, if you search for **NODE1.GDG.G00***, no GDG entries will be found. However, you can find these entries by changing the search to **NODE1.GDG.*G00**.

Note: the LISTC or LISTF prompting panels can be bypassed by using the MASK command (see **MASK Function** on page 166).

Example LISTC

Syntax

```
LISTC [name] [RESET] [PROMPT]
```

Aliases LC, LISTC

Defaults none -- add to the LISTC/LISTF table

Required none

Operands

name one to six alphameric characters, the LISTC/LISTF table name to retrieve.
RESET clear the LISTC/LISTF table before adding any new entries.
PROMPT Prompt for LISTC operands even if a LISTC/LISTF table exists.

LISTC Function

Remarks

The LISTC/LISTF ISPF table is displayed in response to a LISTC command. When you are in a LISTC display, you have many options: you may delete a part of the table, sort the data in different directions, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

There are ten defined formats for LISTC/LISTF. You may select which panels you wish to utilize with the SETPANEL command as shown below and you can switch between these panel formats with **RIGHT** and **LEFT** commands. Following this panel is a sample of each of the ten defined panel formats. Note that StarWarp displays DSORG (under header DO) as DA for direct, PE for partitioned extended, PO for partitioned, PS for sequential and VS for VSAM.

Each of the above panels contain a column of data under the **C** (for Catalog) header which is **Y** if the data set is cataloged, **N** if the data set is not cataloged and **-** if the data set's catalog status was not checked.

The column of data under the V (for Volume) header provides feedback as to whether the data set is present on the volume. **Y** and **N** are short for YES or NO, **M** is short for Managed (meaning that it is a SMS data set) and **-** means that the status of the data set was not checked.

The following primary commands are supported directly for the LISTC function; for documentation on ISPMODE commands available anywhere in StarWarp, see **Common Commands** on page 253.

ALT[ERNAT] displays an alternate view of the LISTC/LISTF table

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

DUA[L] displays a triple line view of the LISTC/LISTF table.

EDITT[BL] (or **ET[BL]**) enters an edit session on LISTC/LISTF table data.

ER[ASE] deletes the LISTC/LISTF table in memory and on disk.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
       [FIRST/LAST/PREV]    [PREFIX/SUFFIX/WORD]
       [ANY/DSNAME/VOLUME/DO/RECFM/CDATE/RDATE]
```

FI[ND] global command, changes to each table data set and issues a FIND subcommand.
Note: in LISTC/LISTF tables, non-partitioned data sets are skipped.
Syntax:
```
FIND memgroup 'anystring'
          [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
          [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
```

GL[OBAL] global command, changes to each data set in the table and issues a StarWarp subcommand.
Syntax: `GLOBAL anysubcommand [anyoperands]`

ID changes the table ID name.
Syntax: `ID name` (where name is a one to six character alphameric name).

LE[FT] rotates through alternate views of the LISTC/LISTF table. PF keys 10 and 22 are normally set to LEFT.

LISTC search for more data sets from a catalog source (after prompting).
Syntax: `LISTC [name] [RESET]` (where name is 1 to 6 alphameric characters)

LISTF search for more data sets from a volume source (after prompting).
Syntax: `LISTF [name] [RESET]` (where name is 1 to 6 alphameric characters)

LO[AD] loads data set names from operating system control blocks.
Syntax: `LOAD APFLST/LNKLST/LPALST [RESET]`

Note: LOAD LNKLST now supports both static and dynamic linklists.

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.

```
Syntax: LOCATE data
             pdsname/pvolume/cdate/rdate/trksize/
              trkfree/extents/blksize/dsorg/alloc
```

MER[GE] merges data from a saved LISTC/LISTF table into the current table.

Syntax: `MERGE name [GROUP/SAMPLE] [RESET]`

(where name is 1 to 6 alphameric characters)

MODEL global command, changes to each data set in the table and issues a MODEL.

NOR[MAL] displays the default view of the LISTC/LISTF table.

NOS[AVE] specifies that the current LISTC/LISTF table should not be saved on disk regardless of the setting of "Automatic save" in SETALL for LISTC/LISTF.

O[PTIONS] provides primary command selection for the LISTC/LISTF function and operand syntax assistance.

OUT[PUT] outputs the LISTC/LISTF table to print or a data set.

Syntax: `OUTPUT [=c / F(ddname)]`

REF[RESH] global command, updates information and status for all lines in the LISTC/LISTF table. Also, REFRESH corrects volume name entries for cataloged data sets.

REM[OVE] trims the LISTC/LISTF table based on a string match.

```
Syntax: REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
             [ANY/DSNAME/VOLUME/DO/RECFM/CDATE/RDATE]
```

REPL[ACE] global command, changes to each data set in the table and issues a REPLACE subcommand. **Note**: in LISTC/LISTF tables, non-partitioned data sets are skipped.

```
Syntax: REPLACE memgroup 'fromstring' 'tostring'
              [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
              [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
              [WRITE/NOWRITE]
```

RESET often used with data set tagging, clears the DATA/MSG field in all table entries.

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the LISTC/LISTF table. PF keys 11 and 23 are normally set to RIGHT.

SAVE creates a permanent table for use in a different STARWARP session.

Syntax: `SAVE [name] [REPLACE/NOREPL]`

(where name is 1 to 6 alphameric characters)

SEEK global command, changes to each data set in the table and looks for a member. Note: in LISTC/LISTF tables, non-partitioned data sets are skipped.

Syntax: `SEEK member`

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.

```
Syntax: SORT [DSNAME/VOLUME/CDATE/RDATE/SIZE/FREE
           /EXTENTS/BLKSIZE/TYPE/ALLOC]
          [ASCEND/DESCEND]
```

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.

Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

WHO[HAS] global command, issues a WHOHAS subcommand for each data set in the table.

X clears the LISTC/LISTF table relative to the cursor position.

Syntax: `X [ABOVE/BELOW/ALL]`

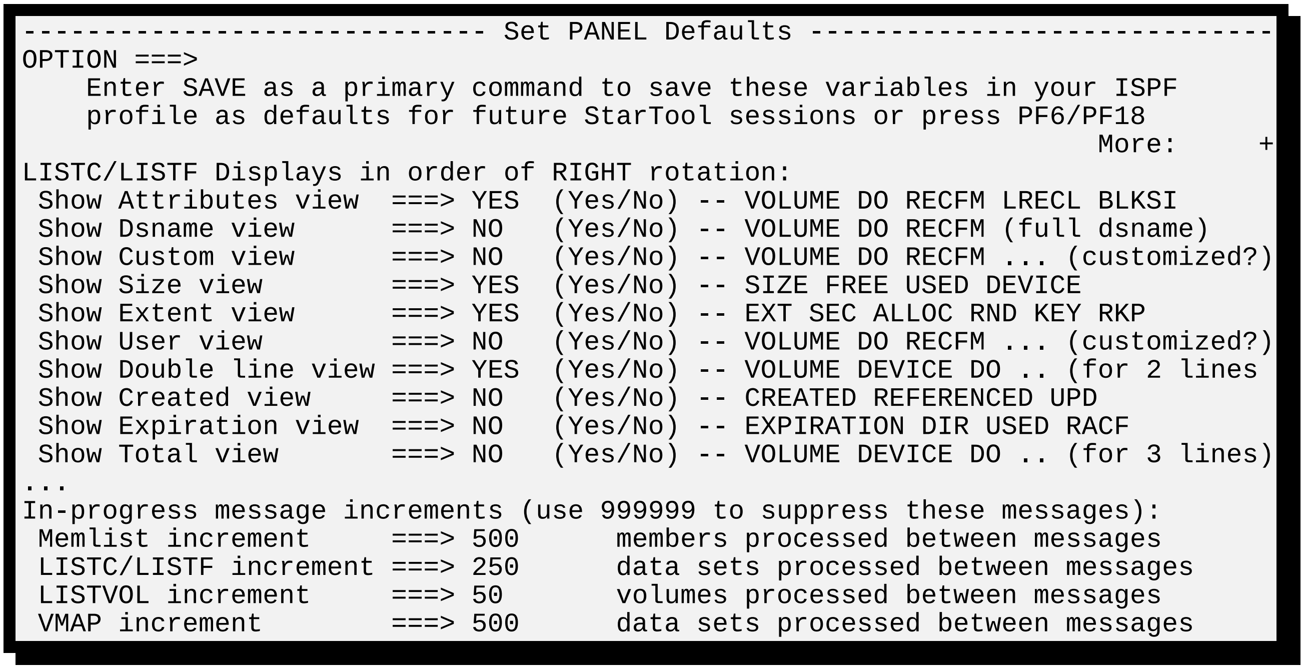
XA[LL] clears the LISTC/LISTF table; this is equivalent to **X ALL**

The following line commands are supported in the LISTC function -- note that the DFHSM commands are supported from the UT panel.

+ provide an extension panel for command entry.
= repeat the previous line command.
6 change to the data set and MEMLIST all.
A change to the data set and execute an alternate subcommand.
ACAT catalog a non-VSAM data set using an alternate user catalog from the UT panel
AUNC uncatalog a non-VSAM data set using an alternate user catalog from the UT panel
B change to the data set and MEMLIST all.
C change to the data set.
CAT catalog this data set.
DEL delete this data set (normally after a confirmation prompt).
E change to the data set and MEMLIST all.
GO change to the data set using GO processing (a number is optional).
HBAC HBBACK DFHSM command to back up the data set
HBDE HBDEL DFHSM command for the data set.
HDEL HDEL DFHSM command for the data set.
HMIG HMIG DFHSM command for the data set.
HML2 HMIG DFHSM command for the data set to LEVEL 2.
HREC HRECALL DFHSM command for the data set.
HSM prompt for various HSM actions on a data set.
IDC perform an IDCAMS LISTC and direct output to the log.
INFO display information on the data set in panel format; also, update data set information.
K kill and clear all following line commands.
LC check the catalog entry for this data set and merge that entry into the LISTC/LISTF table (if different); also, update data set information.
LOG copy the line into the log.
M provide line command selection and entry assistance.
ML change to the data set and MEMLIST all.
MOD change to the data set and MODEL.
O provide line command selection and operand syntax assistance.
PB change to the data set and PBROWSE this data set.
RACF provide entry assist panels for RACF commands: LISTDSD, ADDSD, PERMIT, ALTDSD, DELDSD and SEARCH.
REFR refresh and update information and status for this data set.
REN rename this data set (after prompting).
S select line command (normally CHANGE, it is set by SETSEL).
SEEK change to the data set and check for a member.
SETA save a STARWARP subcommand for repeated use by the ALT line command.
TAG mark this table entry with *TAG* in the DATA/MSG field.
U change to the data set and display USAGE.
UNC uncatalog this data set (not allowed for SMS managed data sets).
UT select the extended user line command panel.
VSAM invoke VSAM manipulation and information services
W add the data set name to the WORKPAD.
WHO check for users of this data set with the WHOHAS subcommand.
X drop the table line.
XCOP change to the data set and interface with extended copy.
XMIT transmit the data set to another user.

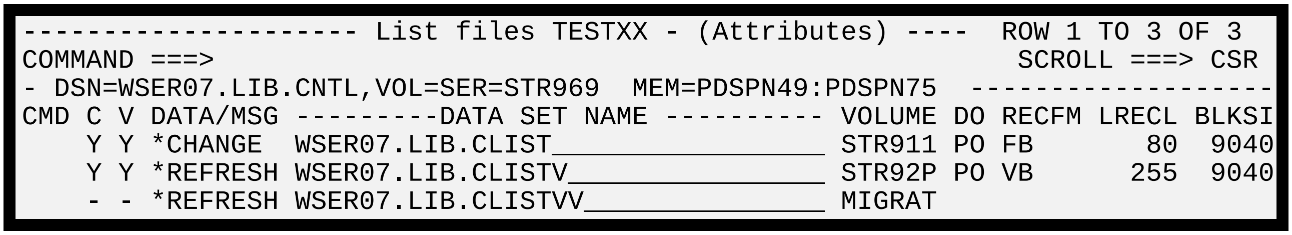
Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTC function:

==	= command, repeat the previous line command for the range of lines.
AA	A command, change and execute an alternate subcommand for each line.
BB	B command, change and MEMLIST all for each line.
CC	C command, change data set for each line.
DDEL	DEL command, delete each data set (normally after a confirmation prompt on each one).
EE	E command, change and MEMLIST all for each line.
GG	GO command, change to each data set using GO processing.
LL	LOG command, copy the range of lines into the log.
MM	ML command, change and MEMLIST all for each line.
OO	O command, provide line command assistance for each line.
SS	S command, select each line in the range of table lines.
SSEE	SEEK command, change to each data set and check for a member.
UU	U command, change and display USAGE for each line.
XX	X command, drop the range of table lines.



```
----------------------------- Set PANEL Defaults -----------------------------
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
                                                                  More:     +
LISTC/LISTF Displays in order of RIGHT rotation:
 Show Attributes view  ===> YES  (Yes/No) -- VOLUME DO RECFM LRECL BLKSI
 Show Dsname view      ===> NO   (Yes/No) -- VOLUME DO RECFM (full dsname)
 Show Custom view      ===> NO   (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Size view        ===> YES  (Yes/No) -- SIZE FREE USED DEVICE
 Show Extent view      ===> YES  (Yes/No) -- EXT SEC ALLOC RND KEY RKP
 Show User view        ===> NO   (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Double line view ===> YES  (Yes/No) -- VOLUME DEVICE DO .. (for 2 lines
 Show Created view     ===> NO   (Yes/No) -- CREATED REFERENCED UPD
 Show Expiration view  ===> NO   (Yes/No) -- EXPIRATION DIR USED RACF
 Show Total view       ===> NO   (Yes/No) -- VOLUME DEVICE DO .. (for 3 lines)
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment     ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment     ===> 50       volumes processed between messages
 VMAP increment        ===> 500      data sets processed between messages
```

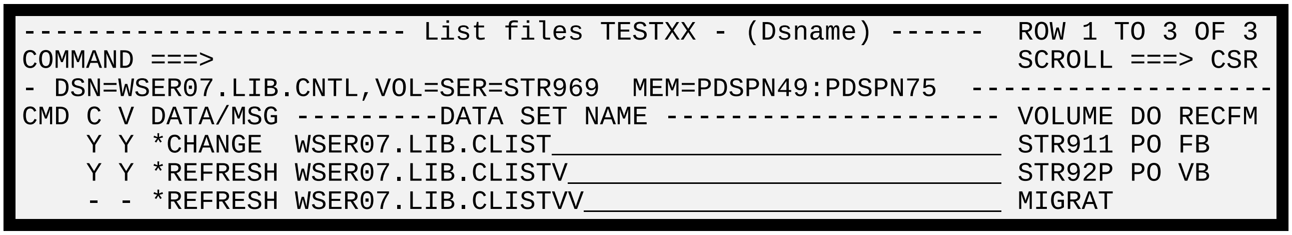
Figure 68. Partial SETPANEL panel with default values for LISTC



```
--------------------- List files TESTXX - (Attributes) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  ------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV________________ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_______________ MIGRAT
```

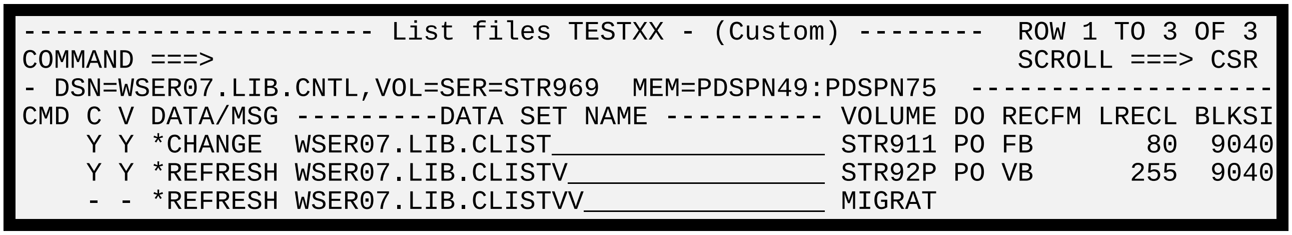
Figure 69. Sample LISTC Table (Attributes view--On by default)

LISTC Function



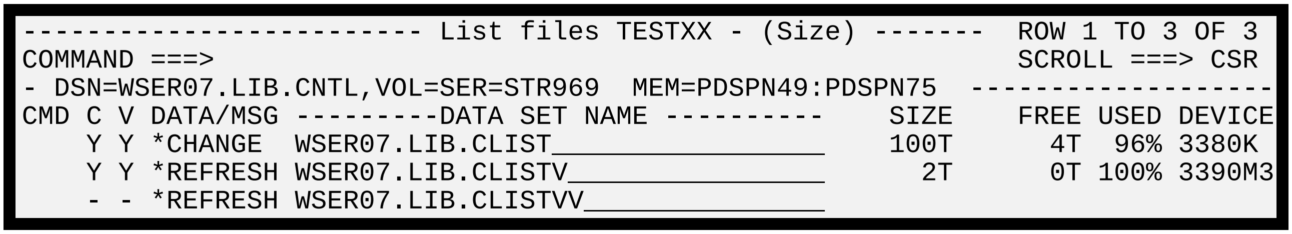
```
------------------------ List files TESTXX - (Dsname) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME --------------------- VOLUME DO RECFM
    Y Y *CHANGE  WSER07.LIB.CLIST____________________________ STR911 PO FB
    Y Y *REFRESH WSER07.LIB.CLISTV___________________________ STR92P PO VB
    - - *REFRESH WSER07.LIB.CLISTVV__________________________ MIGRAT
```

Figure 70. Sample LISTC Table (Dsname view--Off by default)



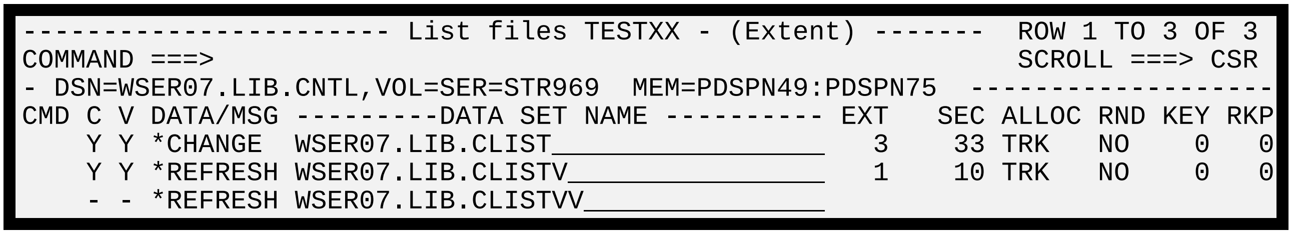
```
---------------------- List files TESTXX - (Custom) --------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV________________ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_______________ MIGRAT
```

Figure 71. Sample LISTC Table (Custom view--Off, can be customized)



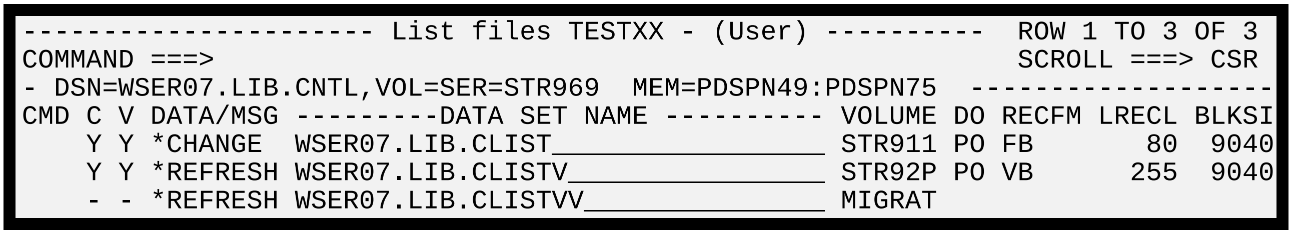
```
------------------------- List files TESTXX - (Size) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ----------    SIZE    FREE USED DEVICE
    Y Y *CHANGE  WSER07.LIB.CLIST_________________    100T      4T  96% 3380K
    Y Y *REFRESH WSER07.LIB.CLISTV________________      2T      0T 100% 3390M3
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

Figure 72. Sample LISTC Table (Size view--On by default)



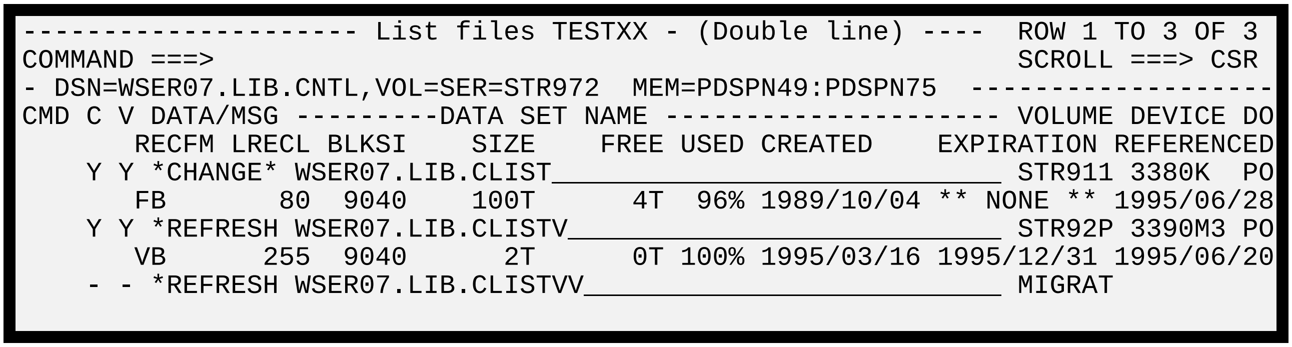
```
----------------------- List files TESTXX - (Extent) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXT   SEC ALLOC RND KEY RKP
    Y Y *CHANGE  WSER07.LIB.CLIST_________________   3    33 TRK   NO    0   0
    Y Y *REFRESH WSER07.LIB.CLISTV________________   1    10 TRK   NO    0   0
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

Figure 73. Sample LISTC Table (Extent view--On by default)



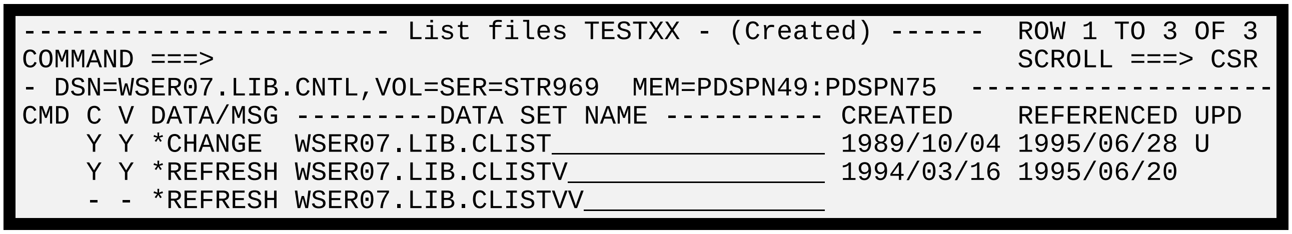
```
---------------------- List files TESTXX - (User) ----------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV________________ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_______________ MIGRAT
```

Figure 74. Sample LISTC Table (User view -- Off, can be customized)



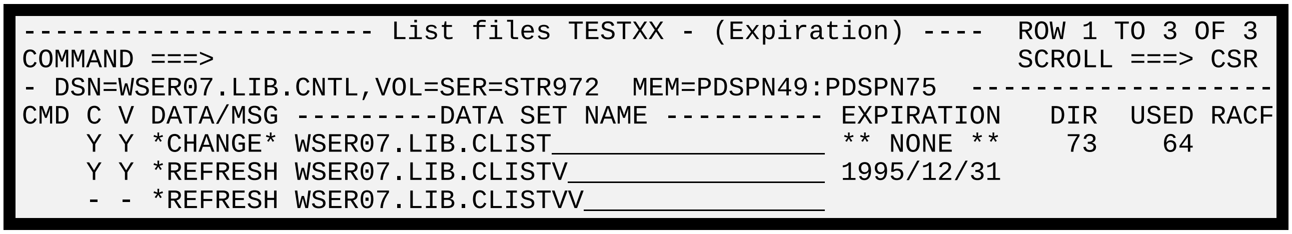
```
--------------------- List files TESTXX - (Double line) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME --------------------- VOLUME DEVICE DO
       RECFM LRECL BLKSI    SIZE    FREE USED CREATED    EXPIRATION REFERENCED
    Y Y *CHANGE* WSER07.LIB.CLIST____________________________ STR911 3380K  PO
       FB       80  9040    100T      4T  96% 1989/10/04 ** NONE ** 1995/06/28
    Y Y *REFRESH WSER07.LIB.CLISTV___________________________ STR92P 3390M3 PO
       VB      255  9040      2T      0T 100% 1995/03/16 1995/12/31 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV__________________________ MIGRAT
```

Figure 75. Sample LISTC Table (Double Line view--On by default)



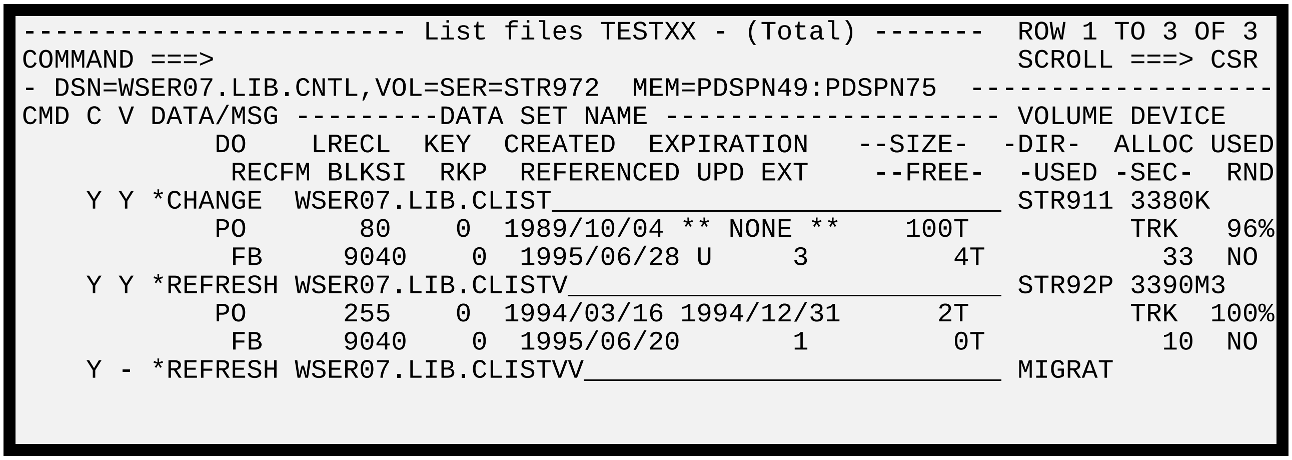
```
----------------------- List files TESTXX - (Created) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- CREATED    REFERENCED UPD
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ 1989/10/04 1995/06/28 U
    Y Y *REFRESH WSER07.LIB.CLISTV________________ 1994/03/16 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

Figure 76. Sample LISTC Table (Created/Referenced view--Off by default)



```
---------------------- List files TESTXX - (Expiration) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXPIRATION   DIR  USED RACF
    Y Y *CHANGE* WSER07.LIB.CLIST_________________ ** NONE **    73    64
    Y Y *REFRESH WSER07.LIB.CLISTV________________ 1995/12/31
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

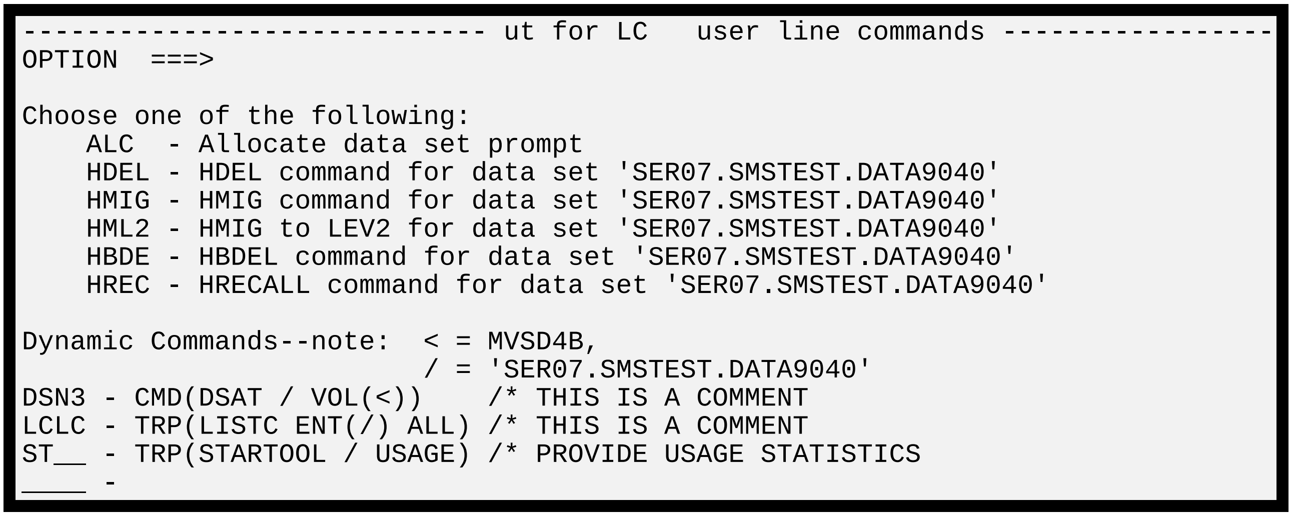
Figure 77. Sample LISTC Table (Expiration view--Off by default)



```
------------------------ List files TESTXX - (Total) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME --------------------- VOLUME DEVICE
            DO    LRECL  KEY  CREATED  EXPIRATION   --SIZE-  -DIR-  ALLOC USED
             RECFM BLKSI  RKP  REFERENCED UPD EXT    --FREE-  -USED -SEC-  RND
    Y Y *CHANGE  WSER07.LIB.CLIST____________________________ STR911 3380K
            PO       80    0  1989/10/04 ** NONE **    100T          TRK   96%
             FB     9040    0  1995/06/28 U     3         4T           33  NO
    Y Y *REFRESH WSER07.LIB.CLISTV___________________________ STR92P 3390M3
            PO      255    0  1994/03/16 1994/12/31      2T          TRK  100%
             FB     9040    0  1995/06/20       1         0T           10  NO
    Y - *REFRESH WSER07.LIB.CLISTVV__________________________ MIGRAT
```

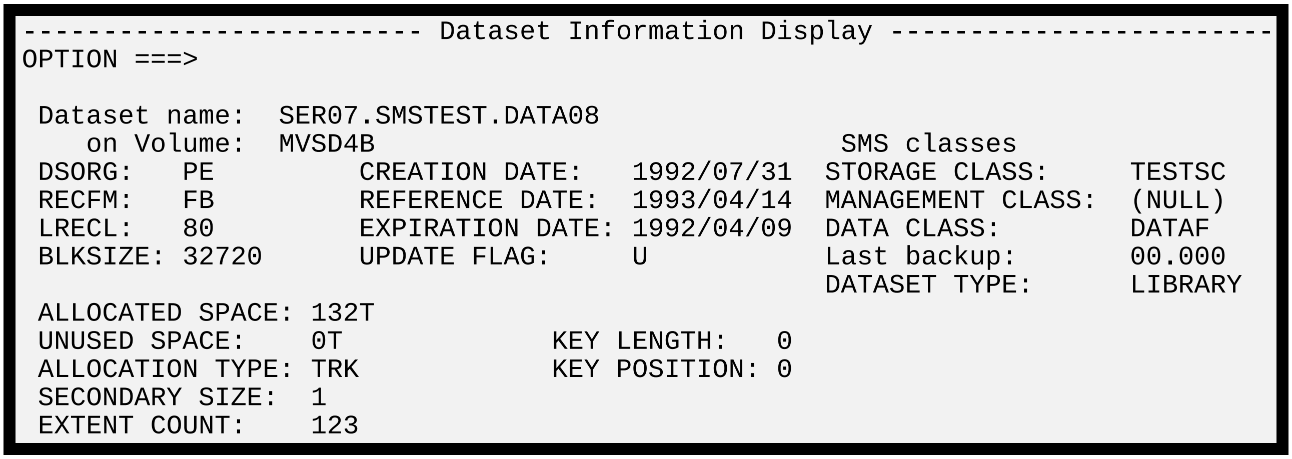
Figure 78. Sample LISTC Table (Total view--Off by default)

LISTC Function



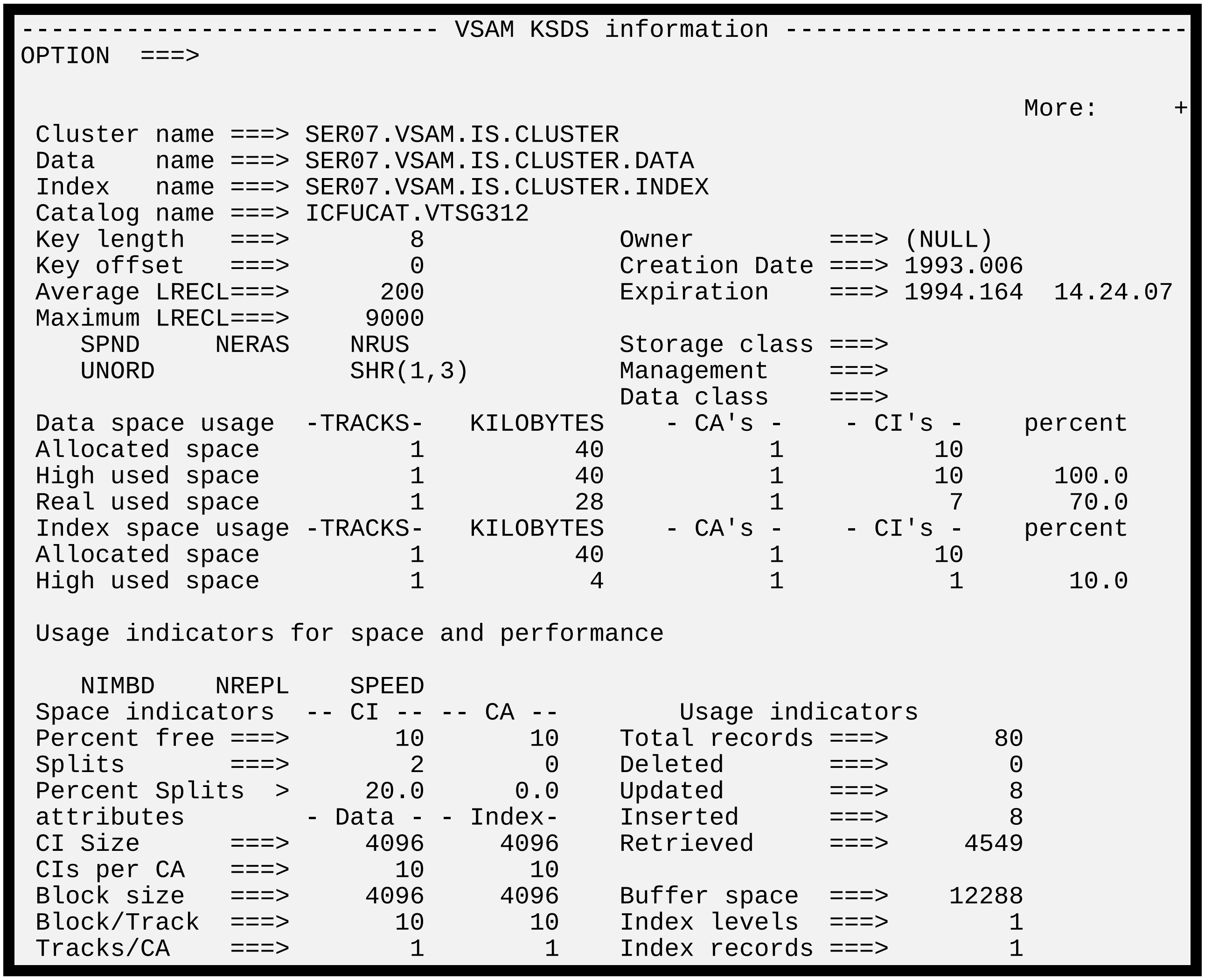
```
----------------------------- ut for LC   user line commands -----------------
OPTION  ===>

Choose one of the following:
    ALC  - Allocate data set prompt
    HDEL - HDEL command for data set 'SER07.SMSTEST.DATA9040'
    HMIG - HMIG command for data set 'SER07.SMSTEST.DATA9040'
    HML2 - HMIG to LEV2 for data set 'SER07.SMSTEST.DATA9040'
    HBDE - HBDEL command for data set 'SER07.SMSTEST.DATA9040'
    HREC - HRECALL command for data set 'SER07.SMSTEST.DATA9040'

Dynamic Commands--note:  < = MVSD4B,
                         / = 'SER07.SMSTEST.DATA9040'
DSN3 - CMD(DSAT / VOL(<))    /* THIS IS A COMMENT
LCLC - TRP(LISTC ENT(/) ALL) /* THIS IS A COMMENT
ST__ - TRP(STARTOOL / USAGE) /* PROVIDE USAGE STATISTICS
____ -
```

Figure 79. Sample LISTC user line command



```
------------------------- Dataset Information Display ------------------------
OPTION ===>

 Dataset name:  SER07.SMSTEST.DATA08
   on Volume:  MVSD4B                              SMS classes
 DSORG:   PE         CREATION DATE:   1992/07/31  STORAGE CLASS:     TESTSC
 RECFM:   FB         REFERENCE DATE:  1993/04/14  MANAGEMENT CLASS:  (NULL)
 LRECL:   80         EXPIRATION DATE: 1992/04/09  DATA CLASS:        DATAF
 BLKSIZE: 32720      UPDATE FLAG:     U           Last backup:       00.000
                                                  DATASET TYPE:      LIBRARY
 ALLOCATED SPACE: 132T
 UNUSED SPACE:    0T             KEY LENGTH:   0
 ALLOCATION TYPE: TRK            KEY POSITION: 0
 SECONDARY SIZE:  1
 EXTENT COUNT:    123
```

Figure 80. Sample LISTC INFO line for a PDSE



```
---------------------------- VSAM KSDS information ---------------------------
OPTION  ===>

                                                                   More:     +
 Cluster name ===> SER07.VSAM.IS.CLUSTER
 Data    name ===> SER07.VSAM.IS.CLUSTER.DATA
 Index   name ===> SER07.VSAM.IS.CLUSTER.INDEX
 Catalog name ===> ICFUCAT.VTSG312
 Key length   ===>        8             Owner          ===> (NULL)
 Key offset   ===>        0             Creation Date ===> 1993.006
 Average LRECL===>      200             Expiration    ===> 1994.164  14.24.07
 Maximum LRECL===>     9000
    SPND     NERAS    NRUS              Storage class ===>
    UNORD             SHR(1,3)          Management    ===>
                                        Data class    ===>
 Data space usage  -TRACKS-   KILOBYTES    - CA's -    - CI's -    percent
 Allocated space          1           40           1          10
 High used space          1           40           1          10       100.0
 Real used space          1           28           1           7        70.0
 Index space usage -TRACKS-   KILOBYTES    - CA's -    - CI's -    percent
 Allocated space          1           40           1          10
 High used space          1            4           1           1        10.0

 Usage indicators for space and performance

    NIMBD    NREPL    SPEED
 Space indicators  -- CI -- -- CA --        Usage indicators
 Percent free ===>       10       10    Total records ===>       80
 Splits       ===>        2        0    Deleted       ===>        0
 Percent Splits  >     20.0      0.0    Updated       ===>        8
 attributes        - Data - - Index-    Inserted      ===>        8
 CI Size      ===>     4096     4096    Retrieved     ===>     4549
 CIs per CA   ===>       10       10
 Block size   ===>     4096     4096    Buffer space  ===>    12288
 Block/Track  ===>       10       10    Index levels  ===>        1
 Tracks/CA    ===>        1        1    Index records ===>        1
```

Figure 81. Sample LISTC INFO line for VSAM

LISTF Function

Purpose The LF or LISTF command is used to display data sets from a volume or VTOC source in an ISPF table. The table displaying LISTF data sets is combined with LISTC data sets.

LISTC/LISTF tables can be saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE, ID, MERGE, NOSAVE and ERASE commands. To automatically save these tables at program end or when a new LISTC/LISTF table is retrieved, check the setting of "Automatic save" in SETALL for LISTC/LISTF.

Saved LISTC/LISTF tables are given members names of the form **PDSLF0na** if the name is one or two numeric digits; otherwise, a name of the form **##name** is used where name is a one to six character alphameric table identification name.

In the LISTF prompt panel, enter a volume mask in the "search" field and the LEVEL/DSN field as a partial data set name. For the LISTF volume mask, **/** may be used for a pattern specification, ***** may be used for a combination name and **:** may be used for a range. For example: **TSO/03**, **TSO*** and **TSO:TSO80**. For the data set name, each level may contain an ***** before or after any characters in that level. For example: **SYS*.NU*.*.*XY**

Note: the LISTC or LISTF prompting panels can be bypassed by using the MASK command (see **MASK Function** on page 166).

Example LISTF

Syntax

```
LISTF [name] [RESET] [PROMPT]
```

Aliases LF, LISTF

Defaults none -- add to the LISTC/LISTF table

Required none

Operands

name	one to six alphameric characters, the LISTC/LISTF table name to retrieve.
RESET	clear the LISTC/LISTF table before adding any new entries.
PROMPT	Prompt for LISTF operands even if a LISTC/LISTF table exists.

Remarks The LISTF/LISTC ISPF table is displayed in response to a LISTF command. When you are in a LISTF display, you have many options: you may delete a part of the table, sort the data in different directions, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

There are ten defined formats for LISTC/LISTF. You may select which panels you wish to utilize with the SETPANEL command as shown below and you can switch between these panel formats with **RIGHT** and **LEFT** commands. Following this panel is a sample of each of the ten defined panel formats. Note that StarWarp displays DSORG (under header DO) as DA for direct, PE for partitioned extended, PO for partitioned, PS for sequential and VS for VSAM.

Each of the above panels contain a column of data under the **C** (for Catalog) header which is **Y** if the data set is cataloged, **N** if the data set is not cataloged and **-** if catalog status was not checked.

The column of data under the **V** (for Volume) header provides feedback as to whether the data set is present on the volume. **Y** and **N** are short for YES or NO, **M** is short for Managed (meaning that it is a SMS data set) and **-** means that the status of the data set was not checked.

The following primary commands are supported directly for the LISTF function; for documentation on ISPMODE commands available anywhere in StarWarp, see **Common Commands,** page 253.

ALT[ERNAT] displays an alternate view of the LISTC/LISTF table

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

DUA[L] displays a triple line view of the LISTC/LISTF table.

EDITT[BL] (or **ET[BL]**) enters an edit session on LISTC/LISTF table data.

ER[ASE] deletes the LISTC/LISTF table in memory and on disk.

EXPR[ESS] executes all entered line commands without pauses between commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
  [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
  [ANY/DSNAME/VOLUME/DO/RECFM/CDATE/RDATE]
```

FI[ND] global command, changes to each table data set and issues a FIND subcommand.
Note: in LISTC/LISTF tables, non-partitioned data sets are skipped.
Syntax:
```
FIND memgroup 'anystring'
       [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
       [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
```

GL[OBAL] global command, changes to each data set in the table and issues a StarWarp subcommand.
Syntax: `GLOBAL anysubcommand [anyoperands]`

ID changes the table ID name.
Syntax: `ID name` (where name is a one to six character alphameric name).

LE[FT] rotates through alternate views of the LISTC/LISTF table. PF keys 10 and 22 are normally set to LEFT.

LISTC search for more data sets from a catalog source (after prompting).
Syntax: `LISTC [name] [RESET]` (where name is 1 to 6 alphameric characters)

LISTF search for more data sets from a volume source (after prompting).
Syntax: `LISTF [name] [RESET]` (where name is 1 to 6 alphameric characters)

LO[AD] loads data set names from operating system control blocks.
Syntax: `LOAD APFLST/LNKLST/LPALST [RESET]`

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax:
```
LOCATE name
          pdsname/pvolume/cdate/rdate/trksize/
           trkfree/extents/blksize/dsorg/alloc
```

MER[GE] merges data from a saved LISTC/LISTF table into the current table.
Syntax: `MERGE name [GROUP/SAMPLE] [RESET]`
(where name is 1 to 6 alphameric characters)

MODEL global command, changes to each data set in the table and issues a MODEL command.

NOR[MAL] displays the default view of the LISTC/LISTF table.

NOS[AVE] specifies that the current LISTC/LISTF table should not be saved on disk regardless of the setting of "Automatic save" in SETALL for LISTC/LISTF.

O[PTIONS] provides primary command selection for the LISTC/LISTF function and operand syntax assistance.

OUT[PUT] outputs the LISTC/LISTF table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REF[RESH] global command, updates information and status for all lines in the LISTC/LISTF table. Also, REFRESH corrects volume name entries for cataloged data sets.

REM[OVE] trims the LISTC/LISTF table based on a string match.
Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
          [ANY/DSNAME/VOLUME/DO/RECFM/CDATE/RDATE]
```

REPL[ACE] global command, changes to each data set in the table and issues a REPLACE.
Note: in LISTC/LISTF tables, non-partitioned data sets are skipped.
Syntax:
```
REPLACE memgroup 'fromstring' 'tostring'
          [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
          [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
          [WRITE/NOWRITE]
```

RESET often used with data set tagging, clears the DATA/MSG field in all table entries.

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the LISTC/LISTF table. PF keys 11 and 23 are normally set to RIGHT.

SAVE creates a permanent table for use in a different STARWARP session.
Syntax: `SAVE [name] [REPLACE/NOREPL]`
(where name is 1 to 6 alphameric characters)

SEEK global command, changes to each data set in the table and looks for a member.
Note: in LISTC/LISTF tables, non-partitioned data sets are skipped.
Syntax: `SEEK member`

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.
Syntax:
```
SORT [DSNAME/VOLUME/CDATE/RDATE/SIZE/FREE
         /EXTENTS/BLKSIZE/TYPE/ALLOC]
        [ASCEND/DESCEND]
```

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

WHO[HAS] global command, issues a WHOHAS subcommand for each data set in the table.

X clears the LISTC/LISTF table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the LISTC/LISTF table; this is equivalent to **X ALL**

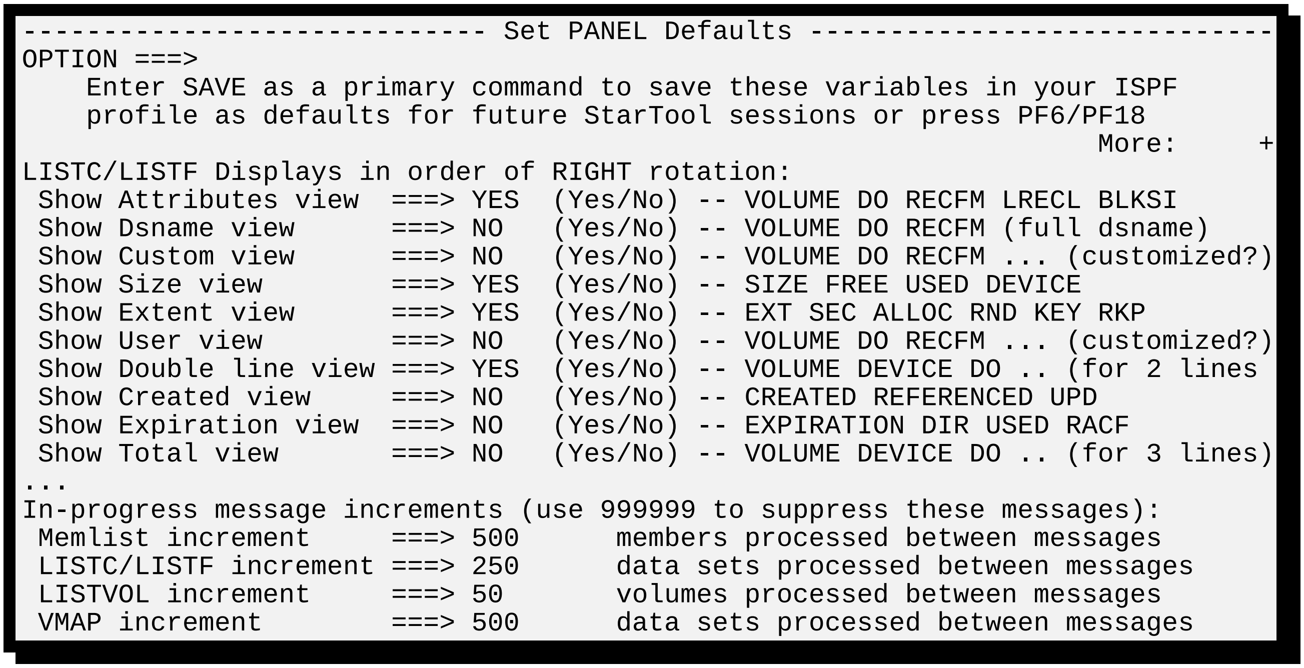
The following line commands are supported in the LISTF function -- note that the DFHSM commands are supported from the UT panel.

+ provide an extension panel for command entry.
= repeat the previous line command.
6 change to the data set and MEMLIST all.
A change to the data set and execute an alternate subcommand.
ACAT catalog a non-VSAM data set using an alternate user catalog from the UT panel
AUNC uncatalog a non-VSAM data set using an alternate user catalog from the UT panel
B change to the data set and MEMLIST all.
C change to the data set.
CAT catalog this data set.

DEL delete this data set (normally after a confirmation prompt).
E change to the data set and MEMLIST all.
GO change to the data set using GO processing (a number is optional).
HBAC HBBACK DFHSM command to back up the data set
HBDE HBDEL DFHSM command for the data set.
HDEL HDEL DFHSM command for the data set.
HMIG HMIG DFHSM command for the data set.
HML2 HMIG DFHSM command for the data set to LEVEL 2.
HREC HRECALL DFHSM command for the data set.
HSM prompt for various HSM actions on a data set.
IDC perform an IDCAMS LISTC and direct output to the log.
INFO display information on the data set in panel format; also, update data set information.
K kill and clear all following line commands.
LC check the catalog entry for this data set and merge that entry into the LISTC/LISTF table (if different); also, update data set information.
LOG copy the line into the log.
M provide line command selection and entry assistance.
ML change to the data set and MEMLIST all.
MOD change to the data set and MODEL.
O provide line command selection and operand syntax assistance.
PB change to the data set and PBROWSE this data set.
RACF provide entry assist panels for RACF commands: LISTDSD, ADDSD, PERMIT, ALTDSD, DELDSD and SEARCH.
REFR refresh and update information and status for this data set.
REN rename this data set (after prompting).
S select line command (normally CHANGE, it is set by SETSEL).
SEEK change to the data set and check for a member.
SETA save a STARWARP subcommand for repeated use by the ALT line command.
TAG mark this table entry with *TAG* in the DATA/MSG field.
U change to the data set and display USAGE.
UNC uncatalog this data set (not allowed for SMS managed data sets).
UT select the extended user line command panel.
VSAM invoke VSAM manipulation and information services
W add the data set name to the WORKPAD.
WHO check for users of this data set with the WHOHAS subcommand.
X drop the table line.
XMIT transmit the data set to another user.

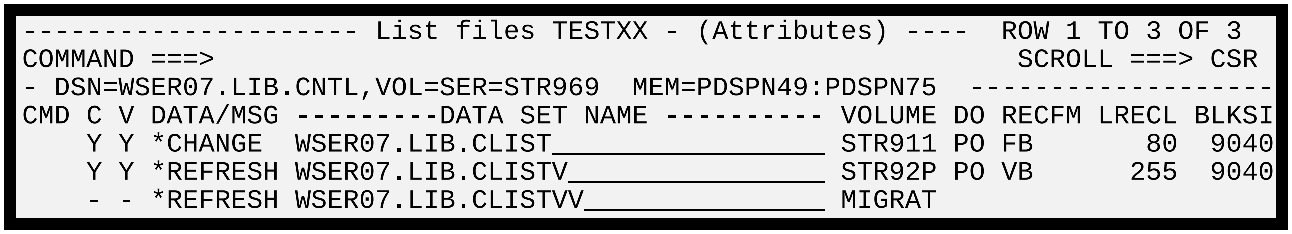
Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in LISTF:
== = command, repeat the previous line command for the range of lines.
AA A command, change and execute an alternate subcommand for each line.
BB B command, change and MEMLIST all for each line.
CC C command, change data set for each line.
DDEL DEL command, delete each data set (normally after a confirmation prompt on each one).
EE E command, change and MEMLIST all for each line.
GG GO command, change to each data set using GO processing.
LL LOG command, copy the range of lines into the log.
MM ML command, change and MEMLIST all for each line.
OO O command, provide line command assistance for each line.
SS S command, select each line in the range of table lines.
SSEE SEEK command, change to each data set and check for a member.
UU U command, change and display USAGE for each line.
XX X command, drop the range of table lines.

LISTF Function



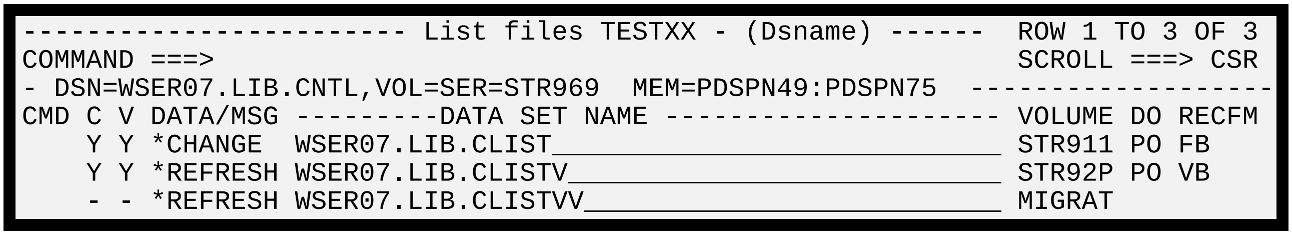
```
----------------------------- Set PANEL Defaults ------------------------------
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
                                                                   More:      +
LISTC/LISTF Displays in order of RIGHT rotation:
 Show Attributes view  ===> YES  (Yes/No) -- VOLUME DO RECFM LRECL BLKSI
 Show Dsname view      ===> NO   (Yes/No) -- VOLUME DO RECFM (full dsname)
 Show Custom view      ===> NO   (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Size view        ===> YES  (Yes/No) -- SIZE FREE USED DEVICE
 Show Extent view      ===> YES  (Yes/No) -- EXT SEC ALLOC RND KEY RKP
 Show User view        ===> NO   (Yes/No) -- VOLUME DO RECFM ... (customized?)
 Show Double line view ===> YES  (Yes/No) -- VOLUME DEVICE DO .. (for 2 lines
 Show Created view     ===> NO   (Yes/No) -- CREATED REFERENCED UPD
 Show Expiration view  ===> NO   (Yes/No) -- EXPIRATION DIR USED RACF
 Show Total view       ===> NO   (Yes/No) -- VOLUME DEVICE DO .. (for 3 lines)
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment     ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment     ===> 50       volumes processed between messages
 VMAP increment        ===> 500      data sets processed between messages
```

Figure 82. Partial SETPANEL panel with default values for LISTF



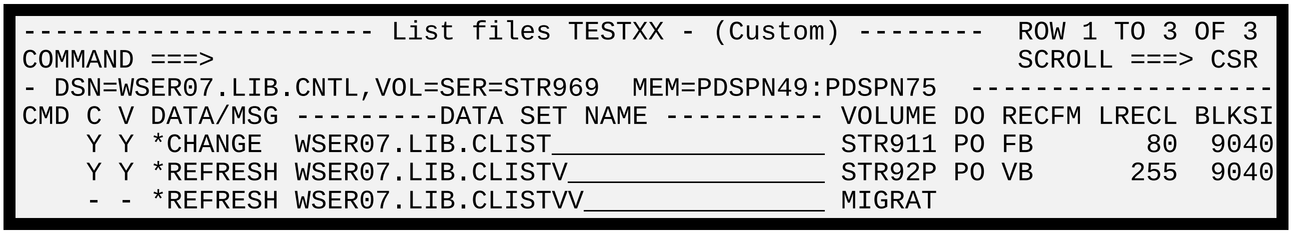
```
--------------------- List files TESTXX - (Attributes) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  --------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV________________ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_______________ MIGRAT
```

Figure 83. Sample LISTF Table (Attributes view--On by default)



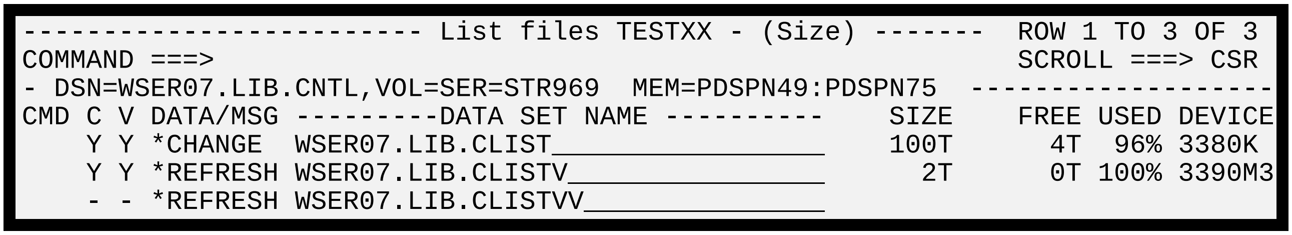
```
------------------------ List files TESTXX - (Dsname) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  --------------------
CMD C V DATA/MSG ---------DATA SET NAME --------------------- VOLUME DO RECFM
    Y Y *CHANGE  WSER07.LIB.CLIST____________________________ STR911 PO FB
    Y Y *REFRESH WSER07.LIB.CLISTV___________________________ STR92P PO VB
    - - *REFRESH WSER07.LIB.CLISTVV__________________________ MIGRAT
```

Figure 84. Sample LISTF Table (Dsname view--Off by default)



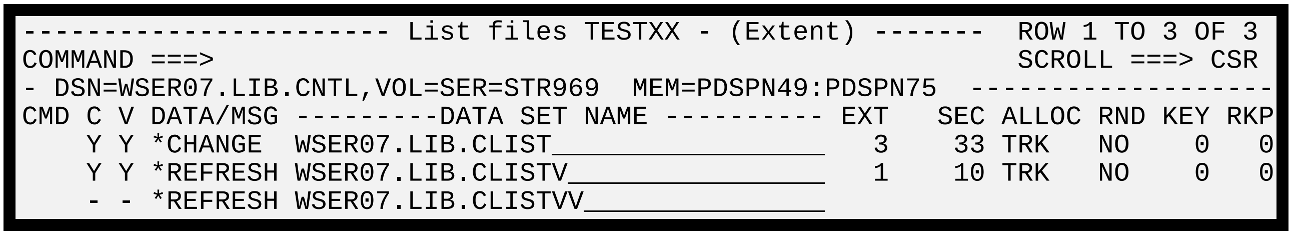
```
---------------------- List files TESTXX - (Custom) --------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  --------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV________________ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_______________ MIGRAT
```

Figure 85. Sample LISTF Table (Custom view--Off, can be customized)



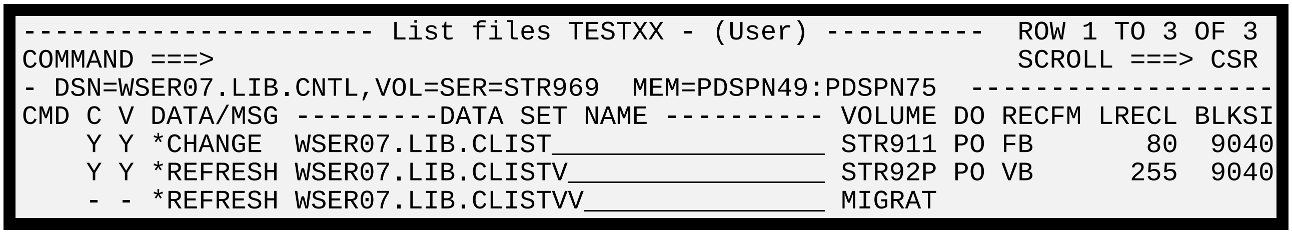
```
------------------------- List files TESTXX - (Size) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ----------    SIZE    FREE USED DEVICE
    Y Y *CHANGE  WSER07.LIB.CLIST_________________    100T      4T  96% 3380K
    Y Y *REFRESH WSER07.LIB.CLISTV________________      2T      0T 100% 3390M3
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

Figure 86. Sample LISTF Table (Size view--On by default)



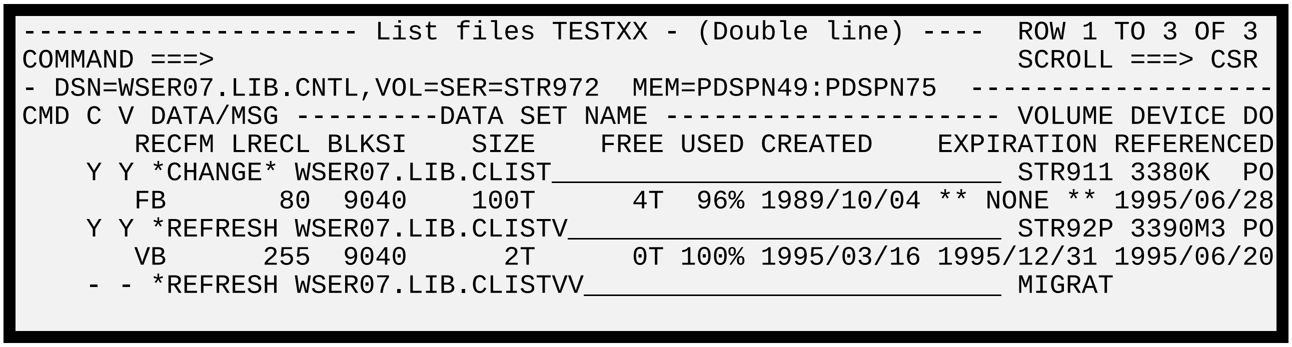
```
----------------------- List files TESTXX - (Extent) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXT   SEC ALLOC RND KEY RKP
    Y Y *CHANGE  WSER07.LIB.CLIST_________________   3    33 TRK   NO    0   0
    Y Y *REFRESH WSER07.LIB.CLISTV________________   1    10 TRK   NO    0   0
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

Figure 87. Sample LISTF Table (Extent view--On by default)



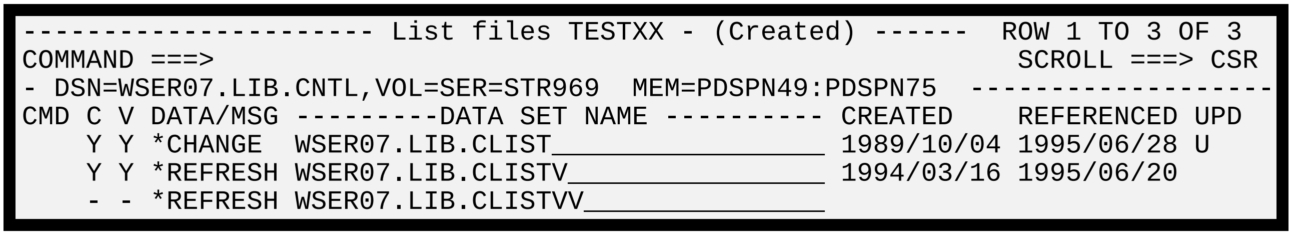
```
---------------------- List files TESTXX - (User) ----------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- VOLUME DO RECFM LRECL BLKSI
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ STR911 PO FB       80  9040
    Y Y *REFRESH WSER07.LIB.CLISTV________________ STR92P PO VB      255  9040
    - - *REFRESH WSER07.LIB.CLISTVV_______________ MIGRAT
```

Figure 88. Sample LISTF Table (User view -- Off, can be customized)



```
--------------------- List files TESTXX - (Double line) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME --------------------- VOLUME DEVICE DO
       RECFM LRECL BLKSI    SIZE    FREE USED CREATED    EXPIRATION REFERENCED
    Y Y *CHANGE* WSER07.LIB.CLIST____________________________ STR911 3380K  PO
       FB       80  9040    100T      4T  96% 1989/10/04 ** NONE ** 1995/06/28
    Y Y *REFRESH WSER07.LIB.CLISTV___________________________ STR92P 3390M3 PO
       VB      255  9040      2T      0T 100% 1995/03/16 1995/12/31 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV__________________________ MIGRAT

```

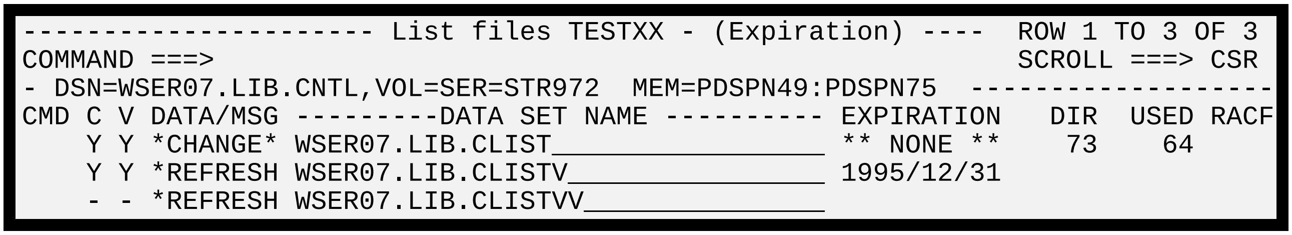
Figure 89. Sample LISTF Table (Double Line view--On by default)



```
---------------------- List files TESTXX - (Created) ------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR969  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- CREATED    REFERENCED UPD
    Y Y *CHANGE  WSER07.LIB.CLIST_________________ 1989/10/04 1995/06/28 U
    Y Y *REFRESH WSER07.LIB.CLISTV________________ 1994/03/16 1995/06/20
    - - *REFRESH WSER07.LIB.CLISTVV_______________
```

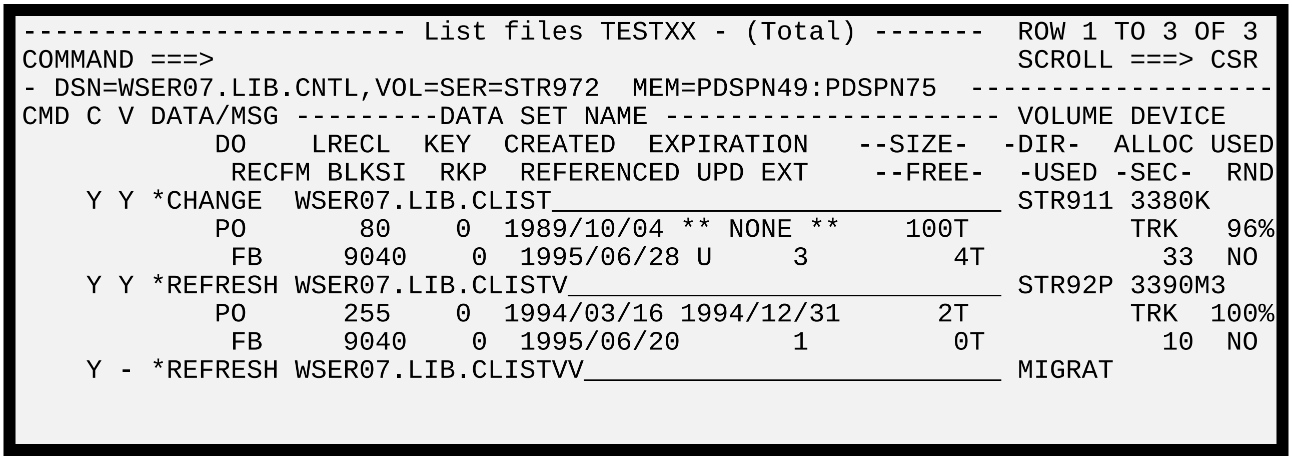
Figure 90. Sample LISTF Table (Created/Referenced view--Off by default)

LISTF Function



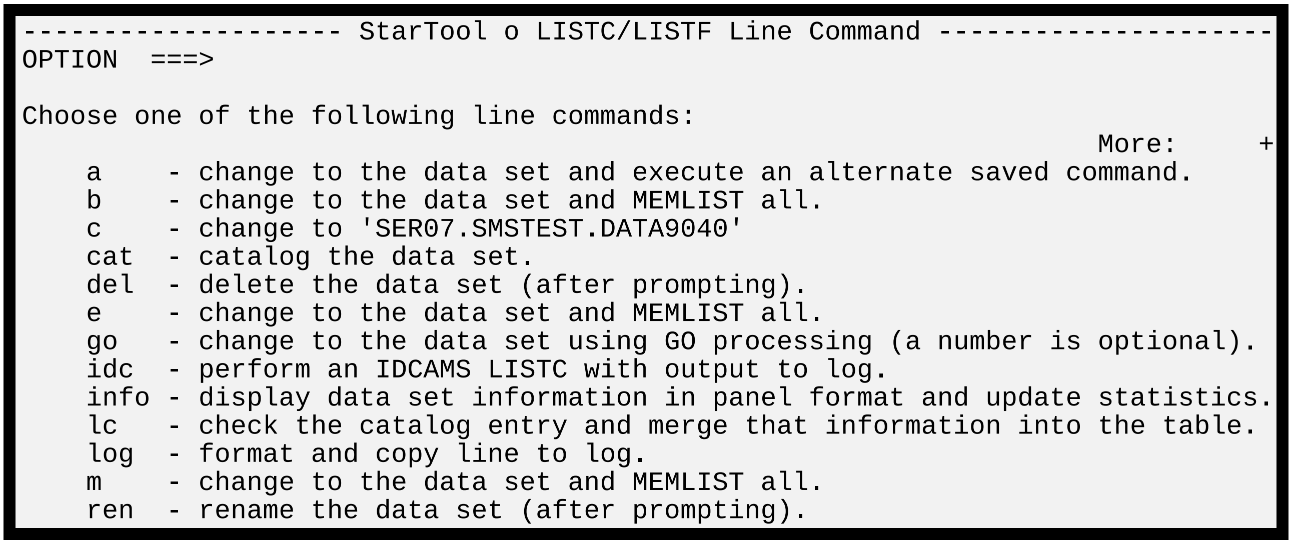
```
---------------------- List files TESTXX - (Expiration) ----  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME ---------- EXPIRATION   DIR  USED RACF
    Y Y *CHANGE* WSER07.LIB.CLIST________________ ** NONE **     73    64
    Y Y *REFRESH WSER07.LIB.CLISTV_______________ 1995/12/31
    - - *REFRESH WSER07.LIB.CLISTVV______________
```

Figure 91. Sample LISTF Table (Expiration view--Off by default)



```
------------------------ List files TESTXX - (Total) -------  ROW 1 TO 3 OF 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=STR972  MEM=PDSPN49:PDSPN75  -------------------
CMD C V DATA/MSG ---------DATA SET NAME --------------------- VOLUME DEVICE
            DO    LRECL  KEY  CREATED  EXPIRATION   --SIZE-  -DIR-  ALLOC USED
             RECFM BLKSI  RKP  REFERENCED UPD EXT    --FREE-  -USED -SEC-  RND
    Y Y *CHANGE  WSER07.LIB.CLIST___________________________ STR911 3380K
            PO       80    0  1989/10/04 ** NONE **    100T         TRK   96%
             FB     9040    0  1995/06/28 U     3         4T            33  NO
    Y Y *REFRESH WSER07.LIB.CLISTV__________________________ STR92P 3390M3
            PO      255    0  1994/03/16 1994/12/31      2T         TRK  100%
             FB     9040    0  1995/06/20       1         0T            10  NO
    Y - *REFRESH WSER07.LIB.CLISTVV_________________________ MIGRAT
```

Figure 92. Sample LISTF Table (Total view--Off by default)



```
-------------------- StarTool o LISTC/LISTF Line Command ---------------------
OPTION  ===>

Choose one of the following line commands:
                                                                   More:     +
    a    - change to the data set and execute an alternate saved command.
    b    - change to the data set and MEMLIST all.
    c    - change to 'SER07.SMSTEST.DATA9040'
    cat  - catalog the data set.
    del  - delete the data set (after prompting).
    e    - change to the data set and MEMLIST all.
    go   - change to the data set using GO processing (a number is optional).
    idc  - perform an IDCAMS LISTC with output to log.
    info - display data set information in panel format and update statistics.
    lc   - check the catalog entry and merge that information into the table.
    log  - format and copy line to log.
    m    - change to the data set and MEMLIST all.
    ren  - rename the data set (after prompting).
```

Figure 93. Sample LISTF line options panel

LISTGRP Subcommand

Purpose The LISTGRP subcommand lists the member group specifications for the current member group.

Note that LISTGRP lists only the names of the member groups; it does not list member names.

Example LISTGRP

Syntax

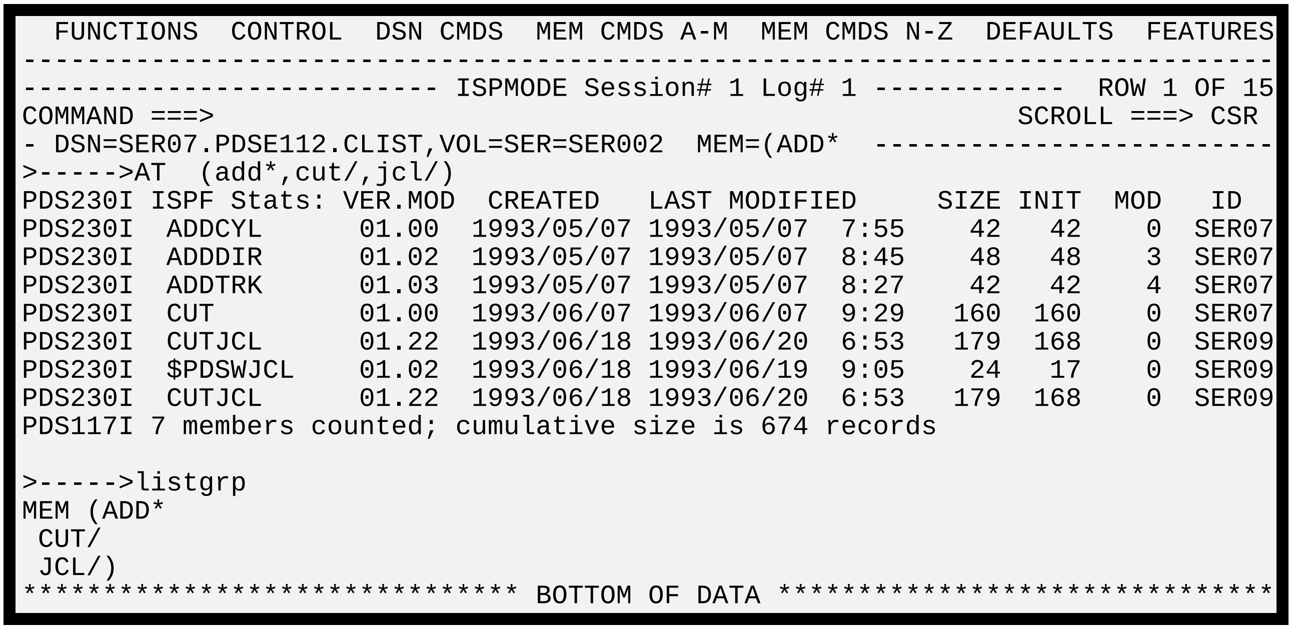
```
LISTGRP
```

Aliases LISTG, LISTGR, LISTGRP

Defaults none

Required none

Operands none



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1 ------------  ROW 1 OF 15
COMMAND ===>                                                 SCROLL ===> CSR
- DSN=SER07.PDSE112.CLIST,VOL=SER=SER002  MEM=(ADD*  ------------------------
>----->AT  (add*,cut/,jcl/)
PDS230I ISPF Stats: VER.MOD  CREATED   LAST MODIFIED     SIZE INIT  MOD   ID
PDS230I  ADDCYL      01.00  1993/05/07 1993/05/07  7:55    42   42    0  SER07
PDS230I  ADDDIR      01.02  1993/05/07 1993/05/07  8:45    48   48    3  SER07
PDS230I  ADDTRK      01.03  1993/05/07 1993/05/07  8:27    42   42    4  SER07
PDS230I  CUT         01.00  1993/06/07 1993/06/07  9:29   160  160    0  SER07
PDS230I  CUTJCL      01.22  1993/06/18 1993/06/20  6:53   179  168    0  SER09
PDS230I  $PDSWJCL    01.02  1993/06/18 1993/06/19  9:05    24   17    0  SER09
PDS230I  CUTJCL      01.22  1993/06/18 1993/06/20  6:53   179  168    0  SER09
PDS117I 7 members counted; cumulative size is 674 records

>----->listgrp
MEM (ADD*
 CUT/
 JCL/)
******************************* BOTTOM OF DATA ******************************
```

Figure 94. Sample LISTGRP Subcommand

LISTV Function

Purpose The LV or LISTV command is used to display disk volume space usage in an ISPF table.

Example LISTV stor:stor03

Syntax

```
LISTV   mask  [RESET] [ADDR/GENERIC]
```

Aliases LV, LISTV

Defaults none -- add to the LISTV table

Required none

Operands

mask if ADDR or GENERIC do not follow this parameter, display all volumes which match the volume mask (mask may contain 1 to 6 characters and ***** or **:** may be used to display all volumes. **/** may be used for a pattern specification, ***** may be used for a combination name and **:** may be used for a range. For example: **TSO/03**,**TSO***,**TSO:TSO80**).

RESET clear the LISTV table before adding any new entries.

ADDR display all volumes which match the partial UCB address (name may contain 1 to 3 characters). For example: **0C**

GENERIC display all volumes which map to the system UNITNAME (name may contain 1 to 8 characters). For example: **SYSDA**

Remarks The LISTV ISPF table is displayed in response to a LISTV command. When you are in a LISTV display, you have many options: you may delete a part of the table, sort the data in different directions, find data in the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

There are five defined formats for LISTV. You may select which panels you wish to utilize with the SETPANEL command as shown below and you can switch between these panel formats with **RIGHT** and **LEFT** commands. Following this panel is a sample of each of the defined formats.

The Attributes LISTV panel shows basic volume space information and volume name, device address and type, the mount attributes (PRIV, PUB or STOR), the volume use count and several status indicators for the volume under the header STAT-INDC:

A	The volume is Allocated.
C	A Catalog on the volume is open; the CAXWA chain is searched to check this.
D	DOS contamination; Indexed VTOC is disabled.
S	The volume is marked SHARED in the UCB.
I	An Indexed VTOC is on the volume.
M	The volume is Managed by SMS.
P	A Page data set on the volume is open.
V	The volume has the VSAM ownership bit on.
R	A CRA (Catalog Recovery Area) is on the volume.

The DATA/MSG field is set to *VTOCERR if the VTOC cannot be input or *RESRV if the volume is reserved. The VTOC Size panel shows VTOC information for systems programmers. This includes mount attributes (PRIV, PUB or STOR), mount status (RESV, PRES or SYSRes), storage group name (for SMS managed volumes), the volume use count (note that it is displayed as *** if it goes over 999), the VTOC CCHH address, VTOC size in tracks, free DSCB count and free VIR count.

The following primary commands are supported directly for the LISTV function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

ADD adds data sets to the LISTF table for all volumes in the LISTV table without prompting.

ALT[ERNAT] displays an alternate view of the LISTV table

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

DUAL switches to a two line view of the LISTV table.

EDITT[BL] (or **ET[BL]**) enters an edit session on LISTV table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring  [ASIS]
    [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
    [ANY/VOLUME/ADDRESS]
```

LE[FT] rotates through alternate views of the LISTV table. PF keys 10 and 22 are normally set to LEFT.

LISTV prompts for LISTV parameters and adds additional volumes to the LISTV table.
Syntax: `LISTV [mask] [RESET] [ADDR/GENERIC]`

LISTVT[OC] adds data sets to the LISTF table for all volumes in the LISTV table after a prompt.

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax: `LOCATE pvolume/pucb/totalfree/largestfree/attrib`

NOR[MAL] displays the default view of the LISTV table.

O[PTIONS] provides primary command selection for the LISTV function and syntax assistance.

OUT[PUT] outputs the LISTV table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REF[RESH] updates information and status for all lines in the table.

REM[OVE] trims the LISTV table based on a string match.
Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
          [ANY/VOLUME/ADDRESS]
```

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the LISTV table. PF keys 11 and 23 are normally set to RIGHT.

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.
Syntax:
```
SORT [VOLUME/ADDRESS/ATTR/TYPE/SPACE/LARGEST]
     [ASCEND/DESCEND]
```

SPA[CE] updates volume space statistics for each volume in the LISTV table.

STATS updates volume attributes for each volume in the LISTV table.

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

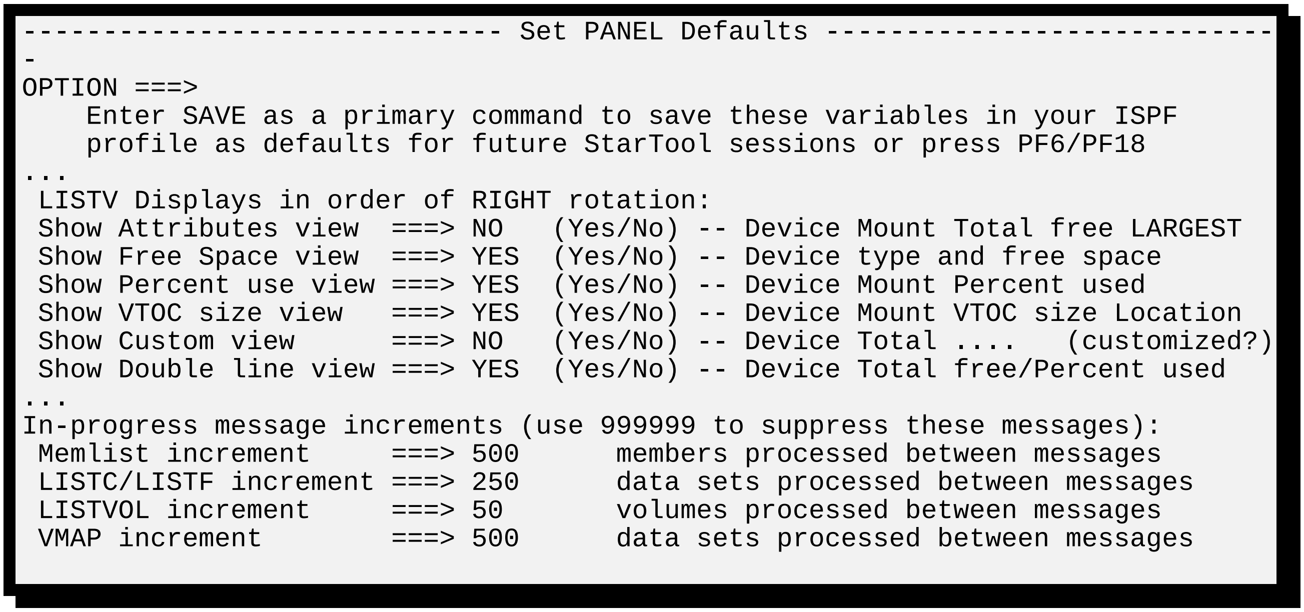
X clears the LISTV table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`
XA[LL] clears the LISTV table; this is equivalent to **X ALL**

The following line commands are supported in the LISTV function:

+ provide an extension panel for command entry.
= repeat the previous line command.
ADD add data sets to the LISTF table for this volume with no prompting.
DIRF tailor batch JCL to set the "DIRF" bit on for this volume.
INDX tailor batch JCL to reverse the indexed VTOC status for this volume (an OS/VTOC will be switched to an indexed status and an indexed volume will be switched to an OS/VTOC)
K kill and clear all following line commands.
LF invoke LISTF to do a VTOC search on this volume after prompting.
LFEX invoke LISTF to do a VTOC search on this volume with no prompting.
LIST add data sets to the LISTF table for this volume after prompting.
LOG copy the line into the log.
M provide line command selection and entry assistance.
O provide line command selection and operand syntax assistance.
REFR update volume space and attributes for this volume.
S select line command (normally SPAC, it is set by SETSEL).
SPAC refresh space data and statistics for this volume.
STAT refresh statistics for this volume.
TAG mark this table entry with *TAG* in the DATA/MSG field.
UT select the extended user line command panel.
VTOC invoke the TSO VTOC command to display volume contents in the log.
VUSE invoke the VUSE subcommand on this volume.
X drop the table line.

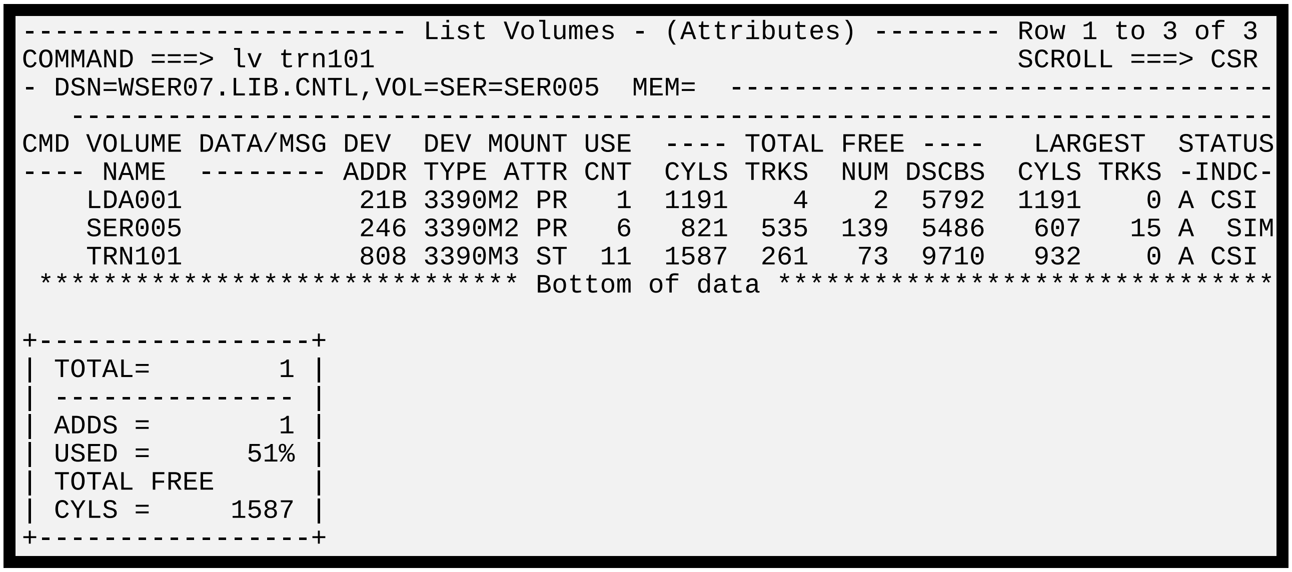
Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the LISTV function:

== = command, repeat the previous line command for the range of lines.
LL LOG command, copy the range of lines into the log.
OO O command, provide line command assistance for each line.
SS S command, select each line in the range of table lines.
XX X command, drop the range of table lines.



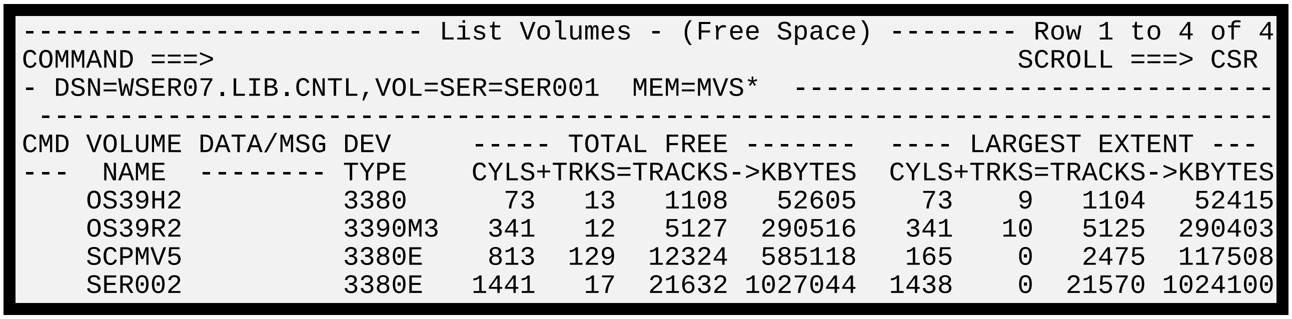
```
------------------------------ Set PANEL Defaults ----------------------------
-
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
...
 LISTV Displays in order of RIGHT rotation:
 Show Attributes view  ===> NO   (Yes/No) -- Device Mount Total free LARGEST
 Show Free Space view  ===> YES  (Yes/No) -- Device type and free space
 Show Percent use view ===> YES  (Yes/No) -- Device Mount Percent used
 Show VTOC size view   ===> YES  (Yes/No) -- Device Mount VTOC size Location
 Show Custom view      ===> NO   (Yes/No) -- Device Total ....   (customized?)
 Show Double line view ===> YES  (Yes/No) -- Device Total free/Percent used
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment     ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment     ===> 50       volumes processed between messages
 VMAP increment        ===> 500      data sets processed between messages
```

Figure 95. Partial SETPANEL panel with defaults for LISTV



```
------------------------ List Volumes - (Attributes) -------- Row 1 to 3 of 3
COMMAND ===> lv trn101                                        SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=  ---------------------------------
   --------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV  DEV MOUNT USE  ---- TOTAL FREE ----   LARGEST  STATUS
---- NAME  -------- ADDR TYPE ATTR CNT  CYLS TRKS  NUM DSCBS  CYLS TRKS -INDC-
    LDA001           21B 3390M2 PR   1  1191    4    2  5792  1191    0 A CSI
    SER005           246 3390M2 PR   6   821  535  139  5486   607   15 A  SIM
    TRN101           808 3390M3 ST  11  1587  261   73  9710   932    0 A CSI
 ****************************** Bottom of data ******************************

+------------------+
| TOTAL=        1 |
| --------------- |
| ADDS =        1 |
| USED =      51% |
| TOTAL FREE      |
| CYLS =     1587 |
+------------------+
```

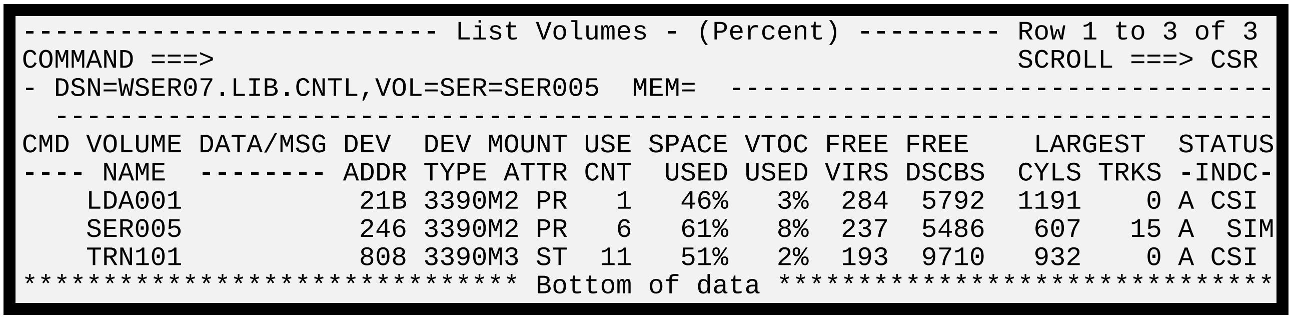
Figure 96. Sample LISTV (Attributes view--Off by default)



```
------------------------ List Volumes - (Free Space) -------- Row 1 to 4 of 4
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=MVS*  -----------------------------
 ----------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV     ----- TOTAL FREE -------  ---- LARGEST EXTENT ---
---  NAME  -------- TYPE    CYLS+TRKS=TRACKS->KBYTES  CYLS+TRKS=TRACKS->KBYTES
    OS39H2          3380      73   13   1108   52605    73    9   1104   52415
    OS39R2          3390M3   341   12   5127  290516   341   10   5125  290403
    SCPMV5          3380E    813  129  12324  585118   165    0   2475  117508
    SER002          3380E   1441   17  21632 1027044  1438    0  21570 1024100
```

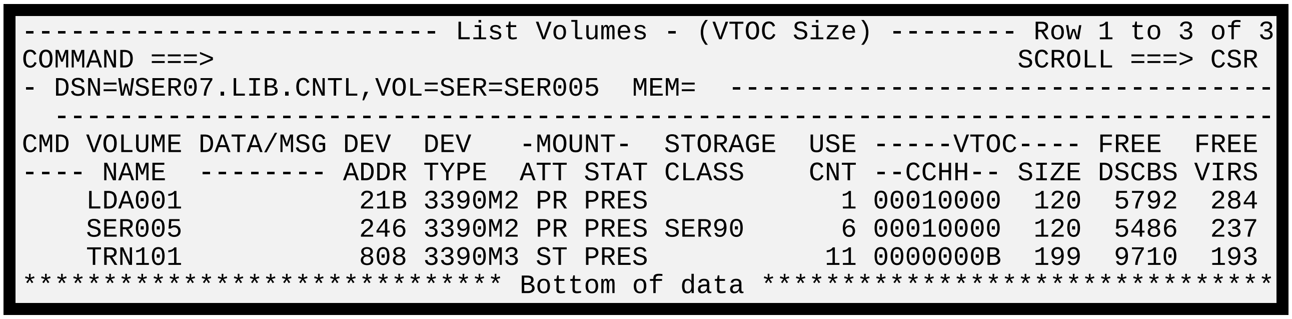
Figure 97. Sample LISTV (Free space view--On by default)

LISTV Function



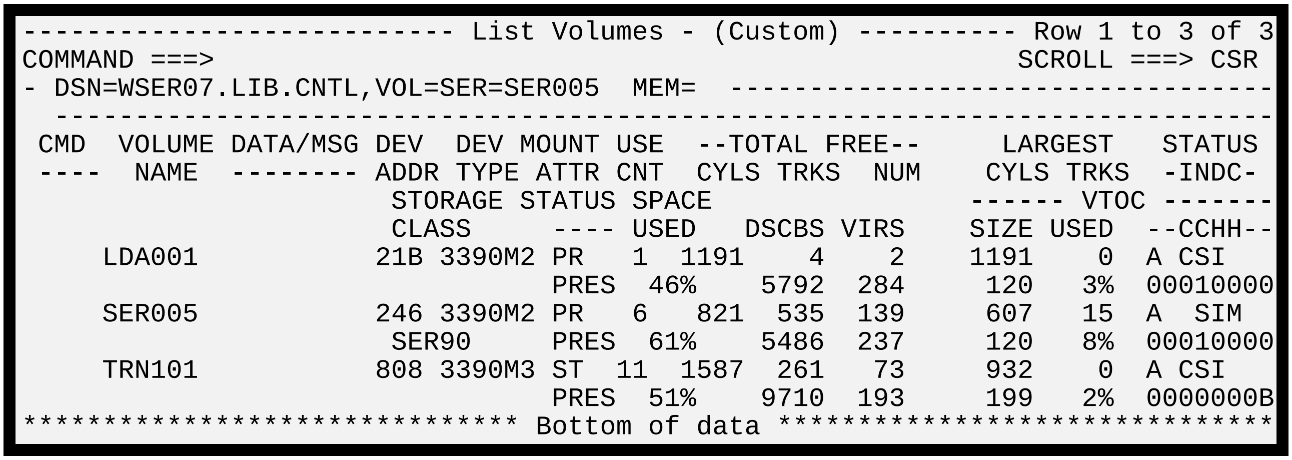
```
-------------------------- List Volumes - (Percent) --------- Row 1 to 3 of 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=  ---------------------------------
  ---------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV  DEV MOUNT USE SPACE VTOC FREE FREE    LARGEST  STATUS
---- NAME  -------- ADDR TYPE ATTR CNT  USED USED VIRS DSCBS  CYLS TRKS -INDC-
    LDA001           21B 3390M2 PR   1   46%   3%  284  5792  1191    0 A CSI
    SER005           246 3390M2 PR   6   61%   8%  237  5486   607   15 A  SIM
    TRN101           808 3390M3 ST  11   51%   2%  193  9710   932    0 A CSI
******************************* Bottom of data ******************************
```

Figure 98. Sample LISTV (Percent view--On by default)



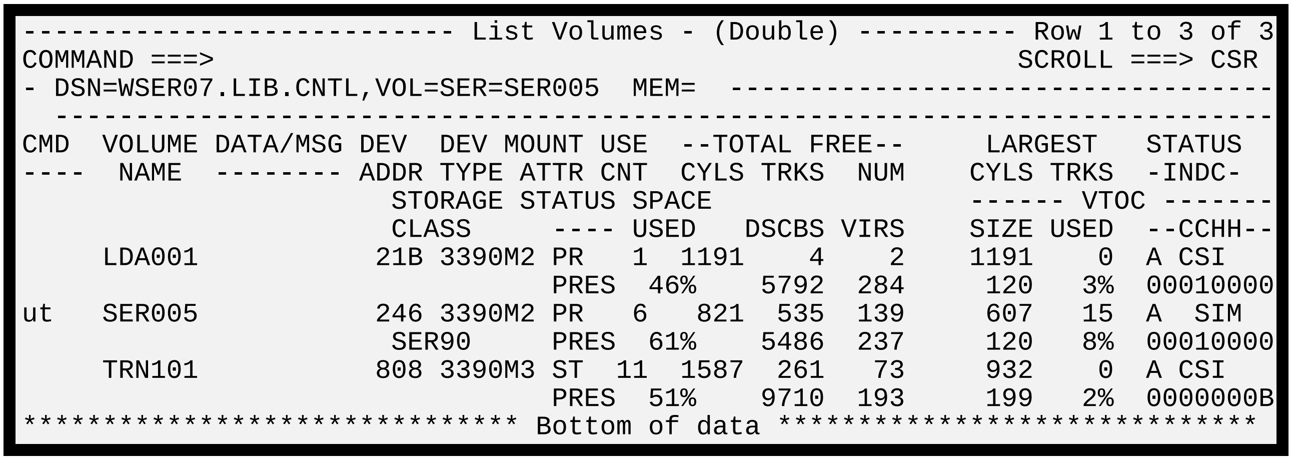
```
-------------------------- List Volumes - (VTOC Size) -------- Row 1 to 3 of 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=  ---------------------------------
  ---------------------------------------------------------------------------
CMD VOLUME DATA/MSG DEV  DEV   -MOUNT-  STORAGE  USE -----VTOC---- FREE  FREE
---- NAME  -------- ADDR TYPE  ATT STAT CLASS    CNT --CCHH-- SIZE DSCBS VIRS
    LDA001           21B 3390M2 PR PRES            1 00010000  120  5792  284
    SER005           246 3390M2 PR PRES SER90      6 00010000  120  5486  237
    TRN101           808 3390M3 ST PRES           11 0000000B  199  9710  193
****************************** Bottom of data *******************************
```

Figure 99. Sample LISTV (VTOC Size view--On by default)



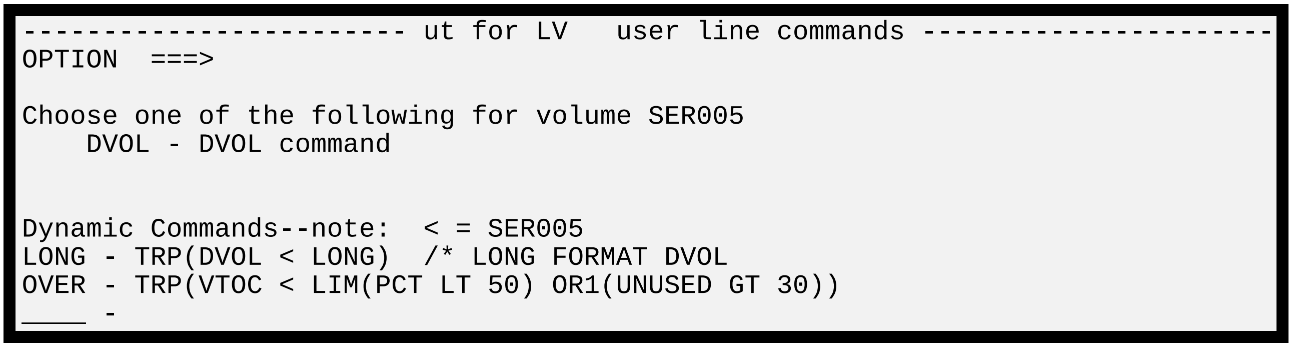
```
--------------------------- List Volumes - (Custom) ---------- Row 1 to 3 of 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=  ---------------------------------
  ---------------------------------------------------------------------------
 CMD  VOLUME DATA/MSG DEV  DEV MOUNT USE  --TOTAL FREE--     LARGEST   STATUS
 ----  NAME  -------- ADDR TYPE ATTR CNT  CYLS TRKS  NUM    CYLS TRKS  -INDC-
                       STORAGE STATUS SPACE                ------ VTOC -------
                       CLASS     ---- USED   DSCBS VIRS    SIZE USED   --CCHH--
     LDA001           21B 3390M2 PR   1  1191    4    2    1191    0  A CSI
                                PRES  46%    5792  284      120   3%  00010000
     SER005           246 3390M2 PR   6   821  535  139     607   15  A  SIM
                       SER90     PRES  61%    5486  237      120   8%  00010000
     TRN101           808 3390M3 ST  11  1587  261   73     932    0  A CSI
                                PRES  51%    9710  193      199   2%  0000000B
******************************* Bottom of data ******************************
```

Figure 100. Sample LISTV (Custom view--Off by default)



```
--------------------------- List Volumes - (Double) ---------- Row 1 to 3 of 3
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER005  MEM=  -----------------------------------
  -----------------------------------------------------------------------------
CMD  VOLUME DATA/MSG DEV  DEV MOUNT USE  --TOTAL FREE--     LARGEST   STATUS
----  NAME  -------- ADDR TYPE ATTR CNT  CYLS TRKS  NUM    CYLS TRKS  -INDC-
                       STORAGE STATUS SPACE                ------ VTOC -------
                       CLASS     ---- USED   DSCBS VIRS    SIZE USED  --CCHH--
     LDA001           21B 3390M2 PR    1  1191    4    2    1191    0  A CSI
                                 PRES  46%    5792  284     120   3%  00010000
ut   SER005           246 3390M2 PR    6   821  535  139     607   15  A  SIM
                       SER90     PRES  61%    5486  237     120   8%  00010000
     TRN101           808 3390M3 ST   11  1587  261   73     932    0  A CSI
                                 PRES  51%    9710  193     199   2%  0000000B
******************************* Bottom of data ******************************
```

Figure 101. Sample LISTV (Double view--On by default)



```
------------------------ ut for LV   user line commands -----------------------
OPTION  ===>

Choose one of the following for volume SER005
    DVOL - DVOL command


Dynamic Commands--note:  < = SER005
LONG - TRP(DVOL < LONG)  /* LONG FORMAT DVOL
OVER - TRP(VTOC < LIM(PCT LT 50) OR1(UNUSED GT 30))
____ -
```

Figure 102. Sample LISTV user line command

LOG Function

Purpose The LOG command displays the current or one of nine session logs. LOG table 1 is the default; it is the table displayed when dialog mode is first started.

Example LOG

Syntax

```
LOG   [num] [RESET] [NODISPL]
```

Aliases LO, LOG, 8

Defaults 1 initially; otherwise, the current log table.

Required none

Operands

num identifies which log table is desired (1 through 9).
RESET empties the selected log table and initializes it with the PDS300I default message.
NODISPL sets the LOG function to use the selected numbered session log; however, it does not cause a switch from the current function. Note: RESET and NODISPL are often used together to clear a log table without actually transiting to the nullified log table.

Remarks The ISPMODE, ISPXEQ or MEMLIST subcommand is used to first enter dialog mode; after you are in dialog mode, you can use the LOG command to display any one of nine log tables.

The log is displayed in response to a LOG command or to display the output from a subcommand. When you are in the log table, you have many options: you may delete a part of the log, find data in the log, print a part of the log, store a part of the log in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as a primary command.

In the following, LOG session processing is described in more detail:

1. The PUTLINE and PUTGET routines are monitored for data.
   a) When the PUTLINE routine is entered, the dialog adds another log table entry.
   b) If an output checkpoint is reached for a subcommand (by default, 250 output lines as set by SETALL for the LOG), ISPMODE positions the log table display to the beginning of the output of this subcommand and allows you to continue or terminate. Note: this screen must be responded to before any STARWARP subcommands may be entered. Some useful commands in this situation are as follows:
      C continue with the output from the subcommand.
      END terminate the subcommand (via a simulated attention).
      F find data in the log.
   c) By default, "CONTROL PROMPT" is enabled. This enables prompting in several situations with STARWARP messages PDS390A through PDS399A at critical decision points in STARWARP processing. STARWARP operates in reply required mode until these messages are responded to and other STARWARP subcommands can not be entered. Some useful commands in this situation are as follows:
      F find data in the log.
      NO indicates that STARWARP should not complete the current action.
      YES indicates that STARWARP should continue with the current action.

d) When the PUTGET routine is entered, the dialog positions the log display to the beginning of the output of the just completed subcommand. Any type of subcommand or command may be entered at this time.

The following primary commands are supported in the LOG function. For documentation on ISPMODE commands available anywhere, see **Common Commands** on page 253.

ALT[ERNAT] displays an alternate view of the log.

BAT[CHJCL] builds batch JCL which will include all STARWARP statements in the LOG table.

COLS provides a column ruler in the log for determining column numbers.

CONT[INUE] after a checkpoint, specifies that STARWARP should continue the current interrupted process until the next checkpoint interval as specified by **SETALL** for the LOG. **END** may be used to terminate the process.

DUA[L] displays a double line view of the log table.

EDITL[OG] (or **EL[OG]**) enters an edit session on the output from the last subcommand.

EDITT[BL] (or **ET[BL]**) enters an edit session on data from the log.

F finds a string; it resets the display start location and positions the cursor.

Syntax:
```
F anystring [nn mm] [ASIS]
   [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
```
(where nn and mm are column numbers)

LE[FT] rotates through alternate views of the log table. PF keys 10 and 22 are normally set to LEFT.

MERGE adds members to the MEMLIST table from a saved MEMLIST table.

Syntax:
```
MERGE  name  [RESET] [NOSTATS]
```
(where name is one to six alphameric characters)

NO after a reply required, specifies that STARWARP should not complete the current action. A **YES** response would allow the process to continue.

NOR[MAL] displays the default view of the log table.

O[PTIONS] provides primary command selection for the log function and operand syntax assistance.

OUT[PUT] outputs the log table to print or a data set.

Syntax:
```
OUTPUT [=c / F(ddname)]
```

RCH[ANGE] captures the command or data set name under the cursor in the log for modification and/or reuse. PF keys 6 and 18 are normally set to RCHANGE.

RF[IND] finds a string (repeat find); it resets the display start location and positions the cursor over the string. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the log table. PF keys 11 and 23 are normally set to RIGHT.

S[ELECT] normally BROWSEs load members and EDITs source members. The value used for the command is set in SETSEL.

Syntax:
```
SELECT member
```

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation. The log UT panel also allows dynamic primary commands with which you specify command names and their corresponding actions.

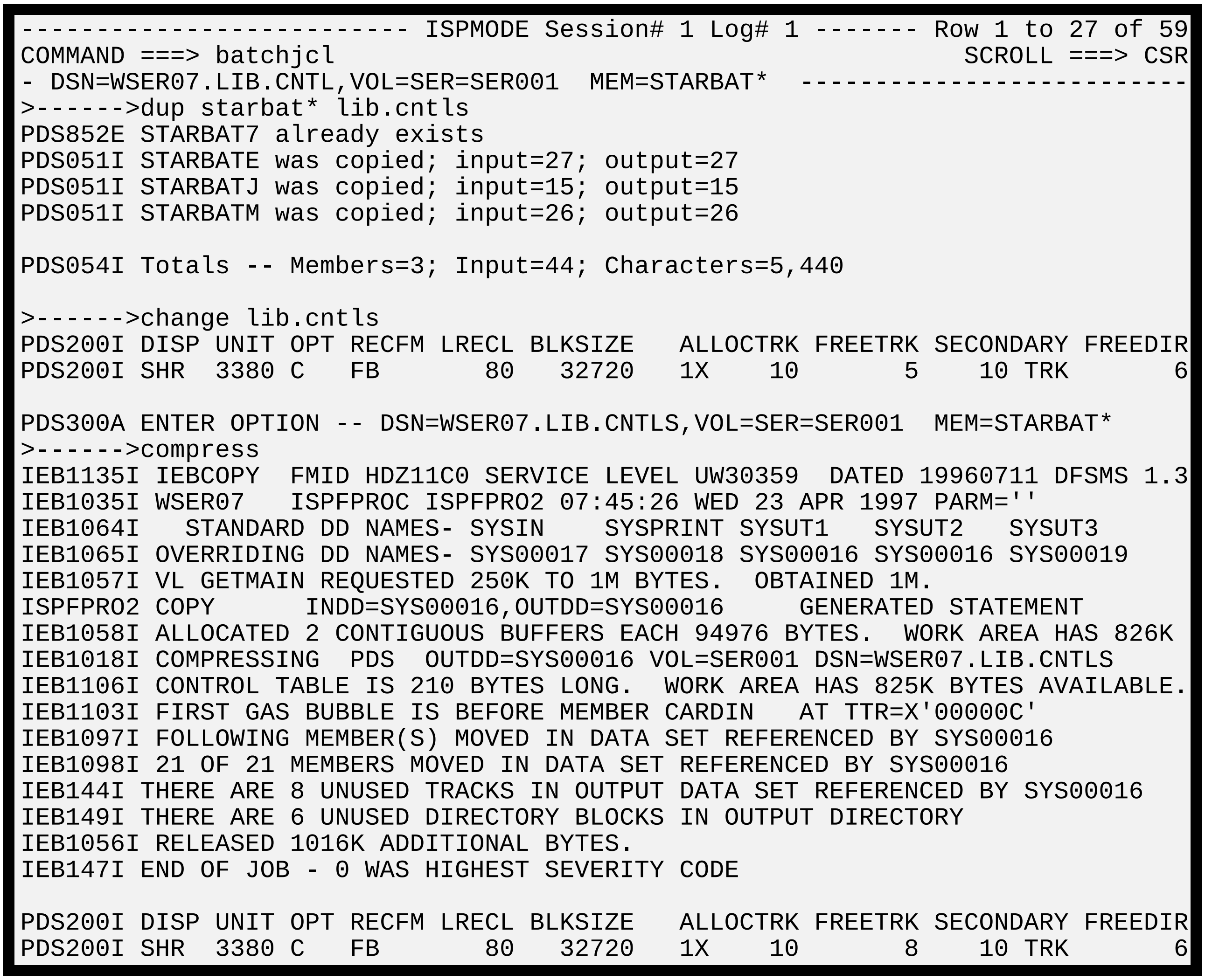
X clears the log table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the log table; this is equivalent to **X ALL**

Y[ES] after a reply required, specifies that STARWARP should continue the current action. A **NO** response would stop the current action.

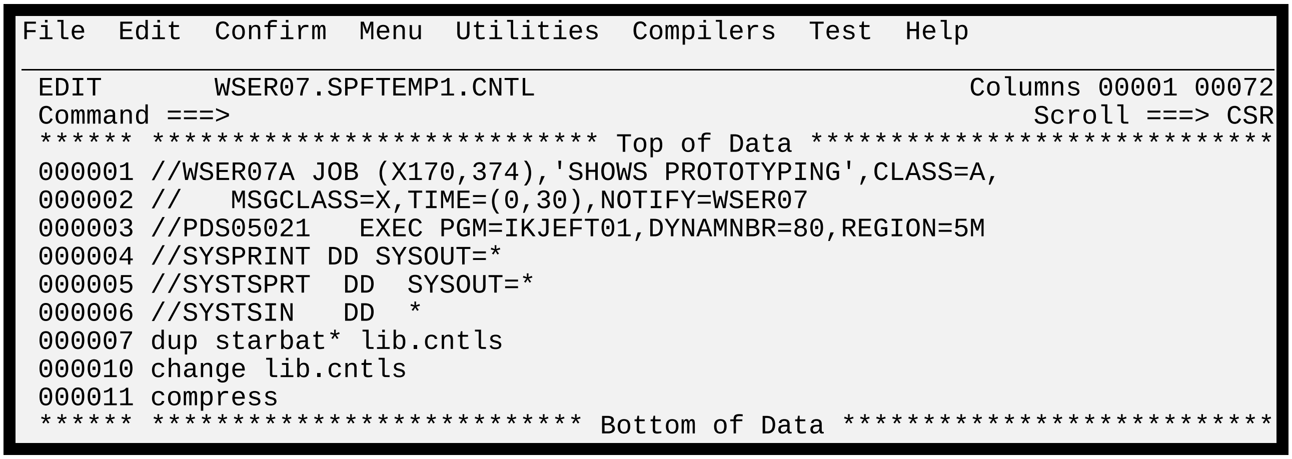
General LOG notes:

1. Only the first 159 characters of an output line are retained in the log; this may be a consideration in LIST, FIND or REPLACE with long lines and list formats NUM, SNUM or NONUM.
2. Echoed subcommand lines are identified with **`>------>`** before the subcommand and any continuation lines are identified by a **-** character in the first log position.
3. The **F** command is a log search command and not the STARWARP FIND subcommand. If you want the FIND subcommand, enter FI, FIN or FIND.
4. Subcommand names PRINT, HELP and TSO are intercepted by ISPF; if you wish to use the STARWARP subcommands with the same names, use an abbreviated name such as PRI, HE or TS.
5. Since output is only displayed at a checkpoint or at the completion of a subcommand, output from a subcommand cannot be monitored as it is produced. Therefore, if STARWARP is in a CPU loop, the screen will lock until an attention is received. At that time, any outputs generated will be displayed and an appropriate response may be given. For further information, see **Appendix D. Attention Processing** on page 269.
6. If a STARWARP subcommand is too long for the 48 character command field, you may enter the first 47 or less characters followed by a **+** to indicate more data is required and a continuation panel with more room for operand entry will be provided.
7. The END command does not work as in STARWARP line mode. END terminates the current function; multiple END commands are normally required to terminate the STARWARP command. Use the QUIT subcommand if you want to terminate STARWARP.



```
------------------------- ISPMODE Session# 1 Log# 1 ------- Row 1 to 27 of 59
COMMAND ===> batchjcl                                             SCROLL ===> CSR
- DSN=WSER07.LIB.CNTL,VOL=SER=SER001  MEM=STARBAT*  -------------------------
>------>dup starbat* lib.cntls
PDS852E STARBAT7 already exists
PDS051I STARBATE was copied; input=27; output=27
PDS051I STARBATJ was copied; input=15; output=15
PDS051I STARBATM was copied; input=26; output=26

PDS054I Totals -- Members=3; Input=44; Characters=5,440

>------>change lib.cntls
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80   32720   1X    10       5    10 TRK       6

PDS300A ENTER OPTION -- DSN=WSER07.LIB.CNTLS,VOL=SER=SER001  MEM=STARBAT*
>------>compress
IEB1135I IEBCOPY  FMID HDZ11C0 SERVICE LEVEL UW30359  DATED 19960711 DFSMS 1.3
IEB1035I WSER07   ISPFPROC ISPFPRO2 07:45:26 WED 23 APR 1997 PARM=''
IEB1064I   STANDARD DD NAMES- SYSIN    SYSPRINT SYSUT1   SYSUT2   SYSUT3
IEB1065I OVERRIDING DD NAMES- SYS00017 SYS00018 SYS00016 SYS00016 SYS00019
IEB1057I VL GETMAIN REQUESTED 250K TO 1M BYTES.  OBTAINED 1M.
ISPFPRO2 COPY      INDD=SYS00016,OUTDD=SYS00016     GENERATED STATEMENT
IEB1058I ALLOCATED 2 CONTIGUOUS BUFFERS EACH 94976 BYTES.  WORK AREA HAS 826K
IEB1018I COMPRESSING  PDS  OUTDD=SYS00016 VOL=SER001 DSN=WSER07.LIB.CNTLS
IEB1106I CONTROL TABLE IS 210 BYTES LONG.  WORK AREA HAS 825K BYTES AVAILABLE.
IEB1103I FIRST GAS BUBBLE IS BEFORE MEMBER CARDIN   AT TTR=X'00000C'
IEB1097I FOLLOWING MEMBER(S) MOVED IN DATA SET REFERENCED BY SYS00016
IEB1098I 21 OF 21 MEMBERS MOVED IN DATA SET REFERENCED BY SYS00016
IEB144I THERE ARE 8 UNUSED TRACKS IN OUTPUT DATA SET REFERENCED BY SYS00016
IEB149I THERE ARE 6 UNUSED DIRECTORY BLOCKS IN OUTPUT DIRECTORY
IEB1056I RELEASED 1016K ADDITIONAL BYTES.
IEB147I END OF JOB - 0 WAS HIGHEST SEVERITY CODE

PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80   32720   1X    10       8    10 TRK       6
```

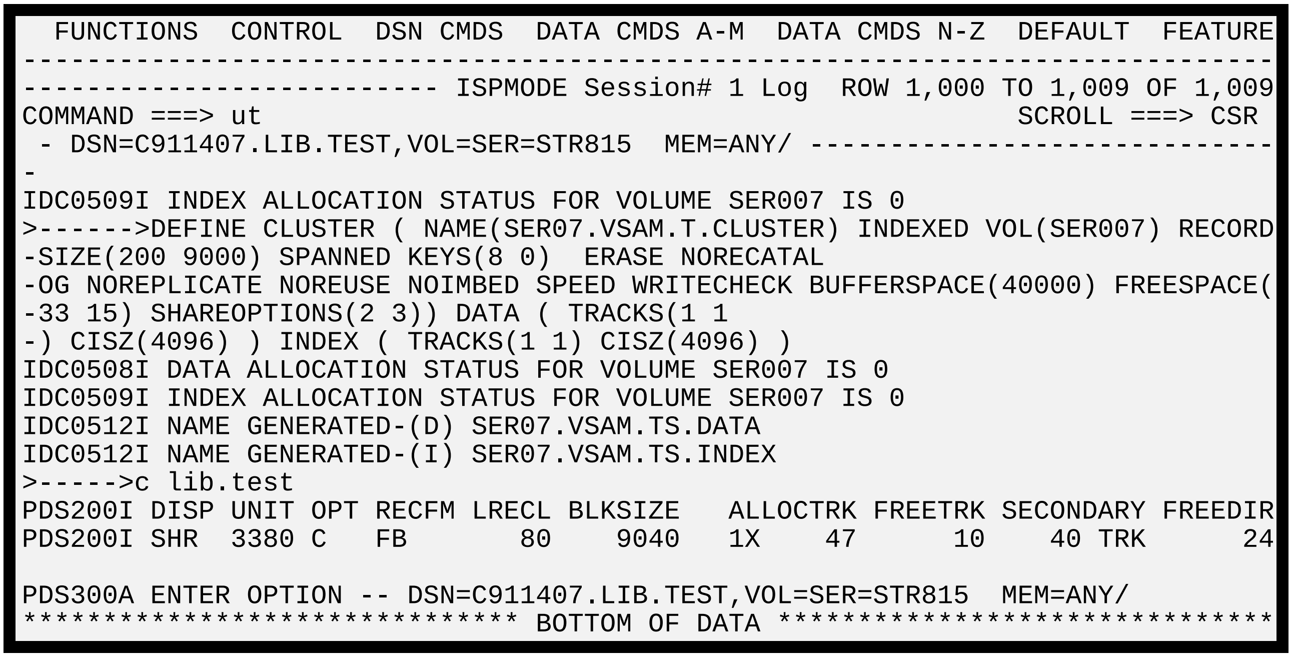
Figure 104. Sample LOG Table (before BATCHJCL)



```
File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help

 EDIT       WSER07.SPFTEMP1.CNTL                           Columns 00001 00072
 Command ===>                                                  Scroll ===> CSR
 ****** **************************** Top of Data *****************************
 000001 //WSER07A JOB (X170,374),'SHOWS PROTOTYPING',CLASS=A,
 000002 //   MSGCLASS=X,TIME=(0,30),NOTIFY=WSER07
 000003 //PDS05021   EXEC PGM=IKJEFT01,DYNAMNBR=80,REGION=5M
 000004 //SYSPRINT DD SYSOUT=*
 000005 //SYSTSPRT  DD  SYSOUT=*
 000006 //SYSTSIN   DD  *
 000007 dup starbat* lib.cntls
 000010 change lib.cntls
 000011 compress
 ****** *************************** Bottom of Data ***************************
```

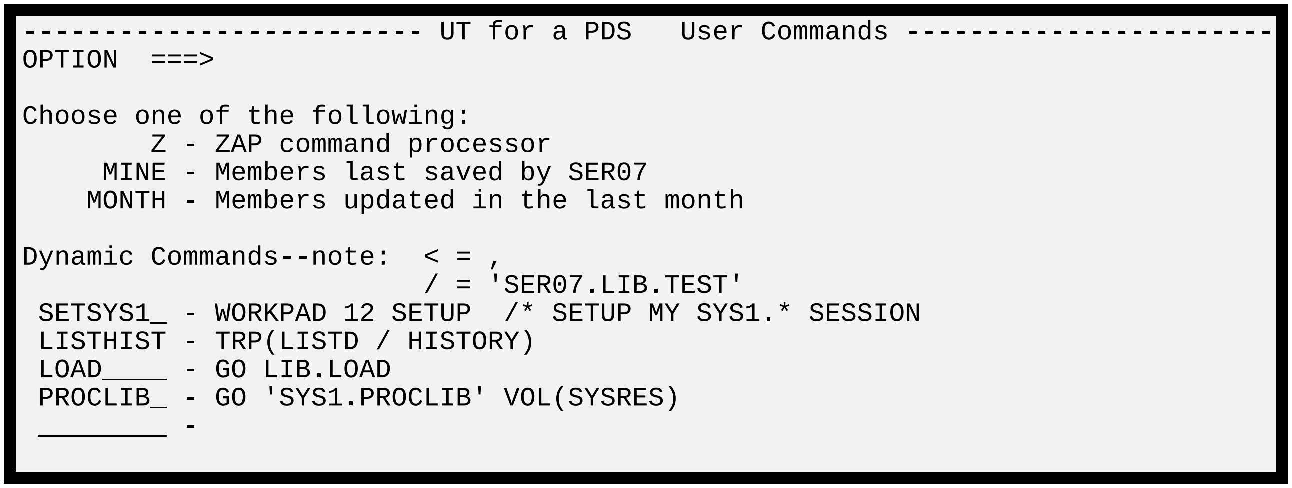
Figure 105. Sample Edit Session (after BATCHJCL)

LOG Function



```
  FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,009 OF 1,009
COMMAND ===> ut                                              SCROLL ===> CSR
 - DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=ANY/ -----------------------------
-
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME SER007 IS 0
>------>DEFINE CLUSTER ( NAME(SER07.VSAM.T.CLUSTER) INDEXED VOL(SER007) RECORD
-SIZE(200 9000) SPANNED KEYS(8 0)  ERASE NORECATAL
-OG NOREPLICATE NOREUSE NOIMBED SPEED WRITECHECK BUFFERSPACE(40000) FREESPACE(
-33 15) SHAREOPTIONS(2 3)) DATA ( TRACKS(1 1
-) CISZ(4096) ) INDEX ( TRACKS(1 1) CISZ(4096) )
IDC0508I DATA ALLOCATION STATUS FOR VOLUME SER007 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME SER007 IS 0
IDC0512I NAME GENERATED-(D) SER07.VSAM.TS.DATA
IDC0512I NAME GENERATED-(I) SER07.VSAM.TS.INDEX
>----->c lib.test
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 C   FB       80    9040   1X    47      10    40 TRK      24

PDS300A ENTER OPTION -- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=ANY/
******************************* BOTTOM OF DATA *******************************
```

Figure 106. Sample LOG Table (two subcommands)



```
------------------------- UT for a PDS   User Commands ----------------------
OPTION  ===>

Choose one of the following:
        Z - ZAP command processor
     MINE - Members last saved by SER07
    MONTH - Members updated in the last month

Dynamic Commands--note:  < = ,
                         / = 'SER07.LIB.TEST'
 SETSYS1_ - WORKPAD 12 SETUP  /* SETUP MY SYS1.* SESSION
 LISTHIST - TRP(LISTD / HISTORY)
 LOAD____ - GO LIB.LOAD
 PROCLIB_ - GO 'SYS1.PROCLIB' VOL(SYSRES)
 ________ -
```

Figure 107. Sample UT primary command panel

MAP Subcommand

Purpose The MAP subcommand lists the CSECT structure of a load module. This map is similar in nature to the output produced by the MAP option of the linkage editor. Note: unresolved external references are also displayed by this subcommand.

Example MAP mema:memb

Syntax

```
MAP   memgroup [FULL/SHORT/ENTRY/RELINK/JCL/LCT                            ]
               [EXTERN/WKEXTERN/PSEUDOREG/LABELREG/COMMON/PRIVATE/CSECT ]
               [OFFSET(hexoff)                                          ]
               [NOSYSTEM                                                ]
               [MODULE({* / Fullm / Partm*)}                            ]
               [REPLACE]
               [MEMBERS / MEMLIST / ML / NEWML / SUBLIST                ]
               [RMODE24/RMODEANY                                        ]
               [AMODE24/AMODE31/AMODEANY                                ]
```

Aliases MA, MAP

Defaults memgroup, FULL

Required none

Operands

memgroup identifies the load member(s) for which map information is desired. Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

FULL provide a full CSECT map; this option also provides the name of the corresponding main module for alias members unless CONTROL NOALIASINFO is in effect.

SHORT provide a CSECT map with the following information omitted: entry names within CSECTS, missing weak external references and alias member names.

ENTRY display only the ENTRY point symbol.

RELINK generate JCL and linkage editor control statements for use in relinking this module.

JCL generate JCL and linkage editor control statements for use in relinking this module.

LCT generate commented linkage attributes and linkage editor control statements for use in managing this module.

OFFSET(hexoff) specifies a 1 to 8 character hexadecimal offset for the origin of this load module.

EXTERN specifies that load modules with missing external symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

WKEXTERN specifies that load modules with missing weak symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

PSEUDOREG specifies that load modules with PSEUDO register symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

LABELREF specifies that load modules with ENTRY symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

COMMON specifies that load modules with COMMON area symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

PRIVATE specifies that load modules with PRIVATE area symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

CSECT specifies that load modules with CSECT symbols are desired. The MODULE keyword may also be specified to search for specific symbols.

MODULE(nm) specifies a 1 to 8 byte partial external name which limits CSECT and ENTRY names for map reporting.

The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.

MODULE(Fullm) report only on CSECT or ENTRY name FULLM.

MODULE(Partm*) report only on CSECT or ENTRY name PARTM...

REPLACE specified with **RELINK**, **LCT** or **JCL** to indicate all CSECTS are to be listed with binder REPLACE statements before the binder INCLUDE statement. **REPLACE** can also be used in conjunction with the **MODULE** keyword to select module names that are to be replaced.

NOSYSTEM specifies that system modules are to be filtered out before reporting MAP information as is done for HISTORY GENERATE. System or compiler routines begin with DFH, DFS, DSN, IBM, IEY, IGY, ILB, ISP, or PLI.

MEMBERS displays the names of members which satisfy the MAP subcommand without changing the current member group.

MEMLIST Same as **ML**. Specifies that any member displayed by the MAP subcommand will be selected for MEMLIST display. The MODULE(name) keyword is used to search for members to display. If no members are selected, a null sublist is the result.

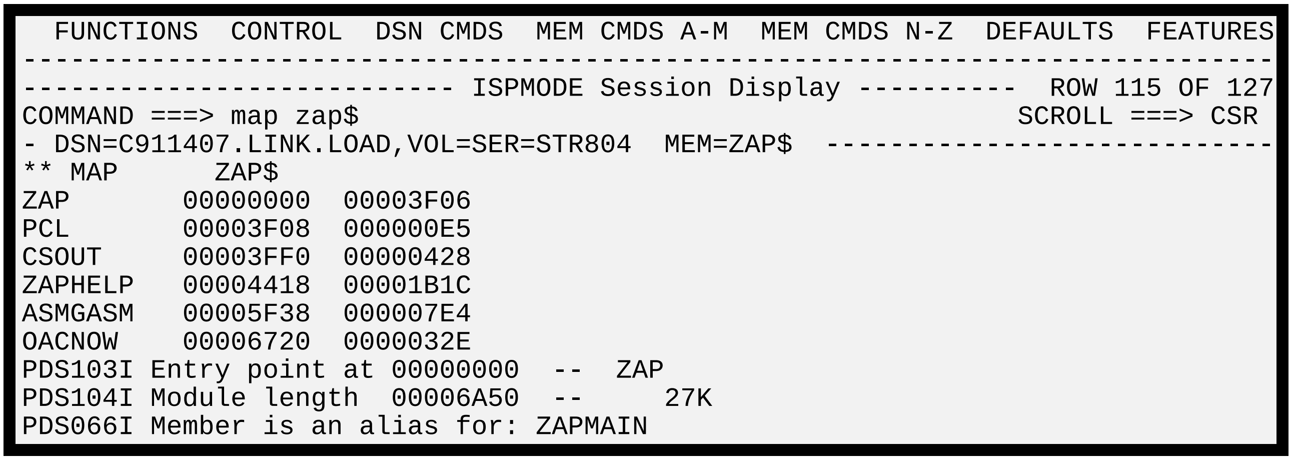
ML same as **MEMLIST**. Specifies that any member displayed by the MAP subcommand will be selected for MEMLIST display. The MODULE(name) keyword is used to search for members to display. If no members are selected, a null sublist is the result.

NEWML Same as **MEMLIST** and **ML** except that the current MEMLIST is reset.

SUBLIST Specifies that any member displayed by the MAP subcommand will be selected for inclusion in a new sublist. The MODULE(name) keyword is used to search for members to display. If no members are selected, a null sublist is the result.

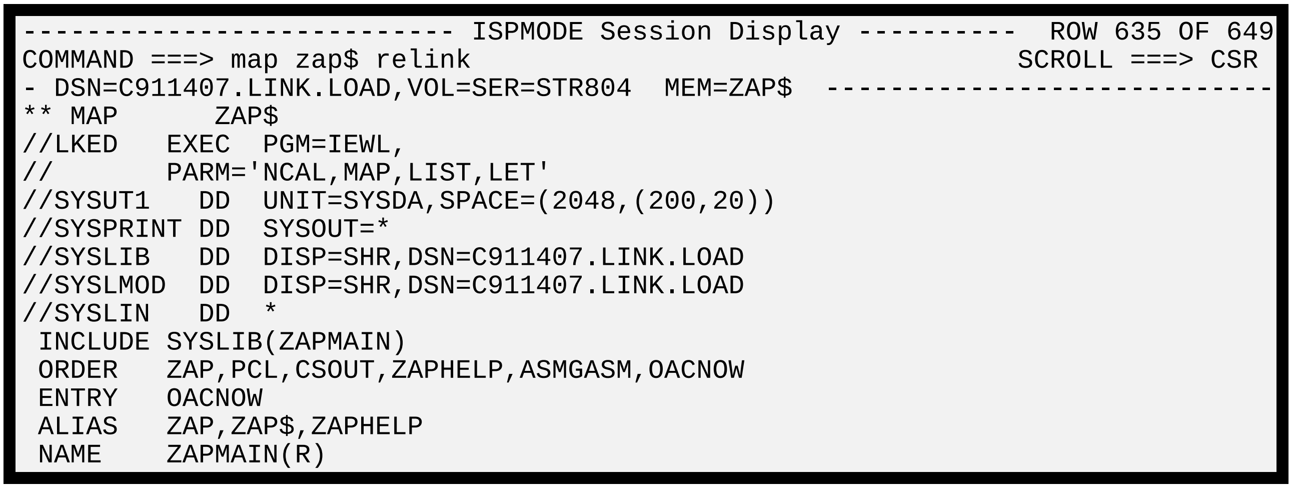
RMODE24/RMODEANY changing of the RMODE at the CSECT level

AMODE24/AMODE31/AMODEANY changing of the AMODE at the CSECT level



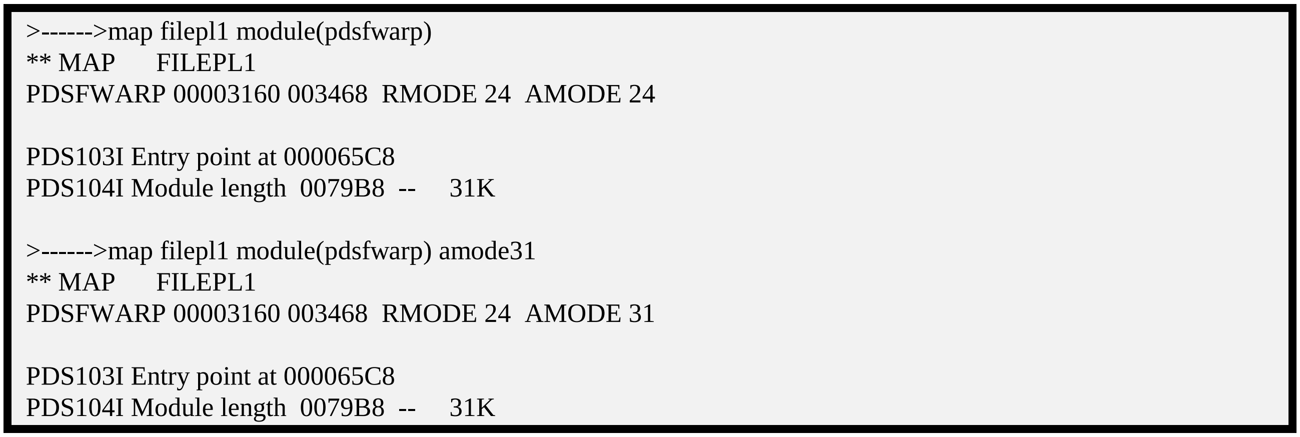
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 115 OF 127
COMMAND ===> map zap$                                            SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAP$  ----------------------------
** MAP      ZAP$
ZAP       00000000  00003F06
PCL       00003F08  000000E5
CSOUT     00003FF0  00000428
ZAPHELP   00004418  00001B1C
ASMGASM   00005F38  000007E4
OACNOW    00006720  0000032E
PDS103I Entry point at 00000000  --  ZAP
PDS104I Module length  00006A50  --     27K
PDS066I Member is an alias for: ZAPMAIN
```

Figure 108. Sample MAP Subcommand



```
--------------------------- ISPMODE Session Display ----------  ROW 635 OF 649
COMMAND ===> map zap$ relink                                     SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=ZAP$  ----------------------------
** MAP      ZAP$
//LKED   EXEC  PGM=IEWL,
//       PARM='NCAL,MAP,LIST,LET'
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(200,20))
//SYSPRINT DD  SYSOUT=*
//SYSLIB   DD  DISP=SHR,DSN=C911407.LINK.LOAD
//SYSLMOD  DD  DISP=SHR,DSN=C911407.LINK.LOAD
//SYSLIN   DD  *
 INCLUDE SYSLIB(ZAPMAIN)
 ORDER   ZAP,PCL,CSOUT,ZAPHELP,ASMGASM,OACNOW
 ENTRY   OACNOW
 ALIAS   ZAP,ZAP$,ZAPHELP
 NAME    ZAPMAIN(R)
```

Figure 109. Sample MAP With RELINK



```
>------>map filepl1 module(pdsfwarp)
** MAP     FILEPL1
PDSFWARP 00003160 003468  RMODE 24  AMODE 24

PDS103I Entry point at 000065C8
PDS104I Module length  0079B8  --    31K

>------>map filepl1 module(pdsfwarp) amode31
** MAP     FILEPL1
PDSFWARP 00003160 003468  RMODE 24  AMODE 31

PDS103I Entry point at 000065C8
PDS104I Module length  0079B8  --    31K
```

Figure xxx. Sample MAP with CSECT mode change

MASK Function

Purpose The MASK command is an extension to the LISTC and LISTF commands. It provides a direct way to enter a data set mask or a volume mask and to then optionally execute the LISTC or LISTF function.

Example MASK ser07.lib.*

Syntax

```
MASK mask [n/VOLUME/VOLUME(vmask)   ]
           [CLEAR/NOCLEAR           ]
           [UPDATE/LC/LISTC/LF/LISTF]
           [RESET                   ]
           [NOFILTER                ]
           [PROMPT                  ]
```

Aliases MAS, MASK

Defaults 1, CLEAR, LISTC

Required mask

Operands

mask identifies the mask which is to be applied to LISTC or LISTF.

For the LISTF volume mask, **/** may be used for a pattern specification, ***** may be used for a combination name and **:** may be used for a range. For example: **TSO/03**, **TSO*** and **TSO:TSO80**.

If the mask is for a data set name, each level may contain an ***** before or after any characters in that level. For example: **SYS1.NU*.*.*XY**. Note that the first level may contain an ***** after a beginning character string.

n/VOLUME identifies the position which this mask is to occupy (1-3 for a data set mask) or VOLUME for a volume mask.

Note that the VOLUME keyword may contain a volume mask; if this is done, the first mask is considered a data set mask ; otherwise, the first mask is considered a volume mask.

CLEAR clear out the other mask entries.
NOCLEAR leave the other mask entries.
UPDATE update the mask only.
LC execute LISTC with this mask.
LISTC execute LISTC with this mask.
LF execute LISTF with this mask.
LISTF execute LISTF with this mask.
RESET clear the current LISTC/LISTF table before executing LISTC or LISTF.
NOFILTER ignores all FILTER options and resets all of the FILTER options to their defaults.
PROMPT display MASK prompt panel after updating with string.

Remarks

MASK provides a way to update LISTC/LISTF parameters without prompting. It could be used from a WORKPAD to build and execute a complicated LISTF table. The following MASK command will build a data set table from a catalog search of data sets using the single data set mask, "USERID.*.CNTL":

```
   MASK USERID.*.CNTL
```

The following MASK command will build a data set table from a VTOC search of all **PUBxxx** volumes using the single data set mask, "USERID.*.CNTL":

```
   MASK USERID.*.CNTL VOLUME(PUB*) LISTF
```

The following MASK commands will build up 3 data set masks. The first MASK command clears all previous masks and the next one adds an additional data set mask. The last MASK command adds a data set mask, a volume mask and starts the VTOC search to build the LISTF table.

```
   MASK SYS1.* 1 UPDATE CLEAR
   MASK SYS2.* 2 UPDATE NOCLEAR
   MASK SYS3.* 3 VOLUME(RES*) NOCLEAR LISTF
```

The above example could also have been specified as follows:

```
   MASK SYS1.* 1 UPDATE CLEAR
   MASK SYS2.* 2 UPDATE NOCLEAR
   MASK SYS3.* 3 UPDATE NOCLEAR
   MASK RES* VOLUME     NOCLEAR LISTF
```

The following MASK command will continue a data set search to additional volumes starting with "SYS":

```
   MASK SYS* VOLUME     NOCLEAR LISTF
```

MEMBERS Subcommand

Purpose The MEMBERS subcommand lists members which are in the current member group.

The MEMBERS subcommand is often used to check member names in a given group before invoking another subcommand which processes member groups.

Example MEMBERS mema:memb

Syntax

```
MEMBERS   memgroup [COUNT / NOCOUNT/ ONLYCOUNT   NOBLDL]
```

Aliases ME, MEM, MEMB, MEMBE, MEMBER, MEMBERS

Defaults memgroup, COUNT

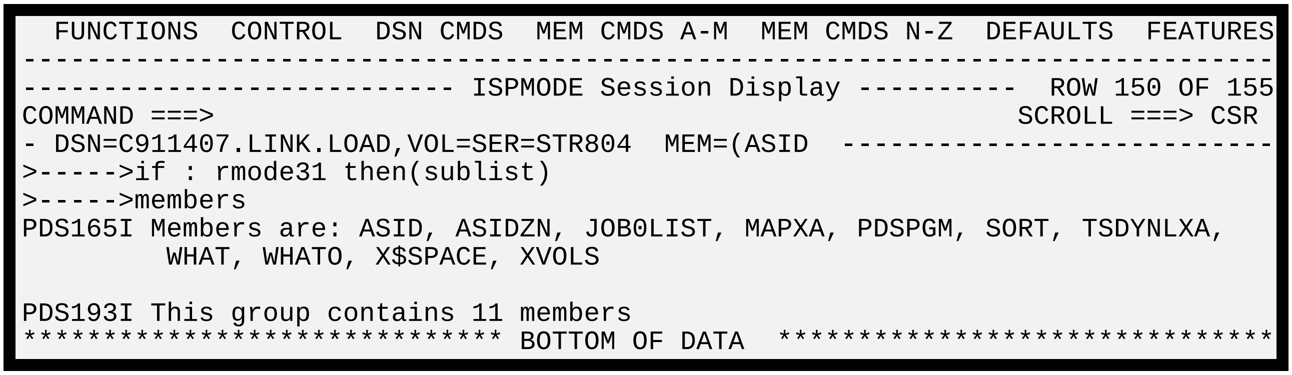
Required none

Operands

memgroup identifies the member(s) whose names are to be displayed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

COUNT issue a summary message to display the number of members in the current group.
NOCOUNT do not issue a summary count message.
ONLYCOUNT issue only a message indicating the number of members in the current group.
NOBLDL specifies that the existence of each member in the current group need not be verified with a BLDL macro. Note: this parameter is intended for high-performance applications in conjunction with the SUBLIST subcommand.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 150 OF 155
COMMAND ===>                                                 SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=(ASID  ---------------------------
>----->if : rmode31 then(sublist)
>----->members
PDS165I Members are: ASID, ASIDZN, JOB0LIST, MAPXA, PDSPGM, SORT, TSDYNLXA,
         WHAT, WHATO, X$SPACE, XVOLS

PDS193I This group contains 11 members
***************************** BOTTOM OF DATA  ******************************
```

Figure 110. Sample MEMBERS Subcommand

MEMLIST Function

Purpose The MEMLIST subcommand builds a scrollable member list display on which STARWARP subcommands can be entered as line commands to process individual members. In addition, ISPF commands, STARWARP subcommands and primary commands may be entered to the right of **COMMAND==>**

Example MEMLIST mema:memb

Syntax

```
MEMLIST memgroup
       [SINCE/BEFORE                                       ]
       [TODAY/YESTERDAY/WEEK/CURRENT/BIWEEK/
           MONTH/QUARTER/HALFYEAR/YEAR/BIYEAR/
           LAST(numdays)/DATE(yyyy/mm/dd)    ]
       [CHANGED(yyyy/mm/dd:yyyy/mm/dd)       ]
       [CREATED(yyyy/mm/dd:yyyy/mm/dd)       ]
       [ABOVE(Count1)                        ]
       [ALIAS/NOALIAS                        ]
       [AMODE24/AMODE31/AMODEANY/NOAMODE24/
          NOAMODE31/NOAMODEANY               ] (load only)
       [AUTH/NOAUTH                          ] (load only)
       [BELOW(Count2)                        ]
       [DC/NODC                              ] (load only)
       [EDIT/NOEDIT                          ] (load only)
       [EXEC/NOEXEC                          ] (load only)
       [FLEVEL/NOFLEVEL                      ] (load only)
       [HASALIAS/NOHASALIAS                  ]
       [ID(Puid)/NOID/NOTID(Puid)            ]
       [LKED(Partl)                          ] (load only)
       [LOADONLY/NOLOADONLY                  ] (load only)
       [MODULE(* / Fullm / Partm*)           ] (load only)
       [NULL/NONULL                          ]
       [ORPHAN/NOORPHAN                      ]
       [OVERLAY/NOOVERLAY                    ] (load only)
       [PAGE/NOPAGE                          ] (load only)
       [REFR/NOREFR                          ] (load only)
       [RENT/NORENT                          ] (load only)
       [REUS/NOREUS                          ] (load only)
       [RMODE24/RMODEANY/NORMODE24/NORMODEANY] (load only)
       [SCTR/NOSCTR                          ] (load only)
       [SSI(hxdata)/SSI/NOSSI/PARTSSI(hxdata)]
       [SYSMOD(Partu) / USERDATA(Partu)      ] (load only)
       [TEST/NOTEST                          ] (load only)
       [TRANS(Partt)                         ] (load only)
       [TTR(Lttr:Httr)                       ]
       [USERDATA(Partu) / SYSMOD(Partu)      ] (load only)
       [USERID(Puid)/NOUSERID/NOTUSERID(Puid)]
       [VSLKED/NOVSLKED                      ] (load only)
       [ZAP(Partz)                           ] (load only)

       [RESET/NORESET                        ]
       [LKEDDATE/NOLKEDDATE                  ] (load only)
```

Aliases ML, MEML, MEMLI, MEMLIS, MEMLIST

Defaults memgroup, NORESET, LKEDDATE, SINCE

Required none

Operands

memgroup identifies the member(s) which are to be added to the member list. Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 261.

SINCE use dates from the specified date to the current date. SINCE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

FROM use dates from the specified date to the current date. FROM indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) follows the date given. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

BEFORE use dates before (and including) the specified date. BEFORE indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

TO use dates before (and including) the specified date. TO indicates that the implied date range (TODAY, YESTERDAY, ... LAST, DATE) is before the given date. For source modules, the date referenced is the ISPF modification date; for load modules, it is the linkage edit date.

TODAY a date, equivalent to LAST(0)

YESTERDAY a date, equivalent to LAST(1)

WEEK a date, equivalent to LAST(7)

CURRENT a date, equivalent to LAST(10)

BIWEEK a date, equivalent to LAST(14)

MONTH a date, equivalent to LAST(30)

QUARTER a date, equivalent to LAST(120)

HALFYEAR a date, equivalent to LAST(183)

YEAR a date, equivalent to LAST(365)

BIYEAR a date, equivalent to LAST(730)

LAST(numdays) a date, indicates the number of days before today.

DATE(cdate) a date, indicates the actual date; it may be entered in ISPF format (yyyy/mm/dd), or Julian format (yyyy.ddd).

CHANGED(date: searches for members modified by an ISPF editor or a ZAP program between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like:
CHANGED(yyyy/mm/dd:yyyy/mm/dd)

Note that only members with ISPF statistics or members created by a linkage editor are supported.

CREATED(date: searches for members created with ISPF statistics or by the linkage editor between the dates specified. If only a single date is entered, only that date is checked but normally a date range is entered like:
CREATED(yyyy/mm/dd:yyyy/mm/dd)

ABOVE(Count1) select members with more than the number of lines (amount of storage for load modules) defined by the number, Count1.

ALIAS select alias members.

NOALIAS select main members.

AMODE24 select modules with addressing mode 24.

NOAMODE24 select modules with addressing mode 31 or ANY.

AMODE31 select modules with addressing mode 31.

NOAMODE31 select modules with addressing mode 24 or ANY.

AMODEANY select modules with addressing mode ANY.

NOAMODEANY select modules with addressing mode 24 or 31.

AUTH select APF authorized modules.

NOAUTH select non-authorized modules.

BELOW(Count2) select members with less than the number of lines (amount of storage for load modules) defined by the number, Count2.

DC select downward-compatible modules.

NODC select no downward-compatible modules.

EDIT select modules which can be reprocessed by the linkage editor.

NOEDIT select modules which can not be reprocessed by the linkage editor.

EXEC select EXECUTABLE modules.

NOEXEC select non-EXECUTABLE modules.

FLEVEL select modules processed by the F-level linkage editor.

NOFLEVEL select modules not processed by the F-level linkage editor.

HASALIAS select main members which have one or more aliases.

NOHASALIAS select alias members or main members without any aliases.

ID(Puid) select members with ISPF statistics and userids matching the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. ID is an alias of USERID.

NOID select members without ISPF statistics. NOID is an alias of NOUSERID.

NOTID(Puid) same as NOTUSERID. select members without ISPF statistics or members with ISPF statistics and userids which do not match the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters.

LKED(Partl) select modules identified as linked by the linkage editor defined by the partial name, **Partl**. **Partl** may be entered as a partial linkage editor IDR name with one to ten characters.

LOADONLY select modules marked for LOAD ONLY.

NOLOADONLY select modules not marked for LOAD ONLY.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT names for attribute searches. Note that if this parameter is entered with any TRANS, SYSMOD/USERDATA or ZAP keywords, only those CSECTs which satisfy the MODULE keyword are checked for the other keyword conditions. The MODULE operand has several valid forms:

- **MODULE(*)** use the last name entered on a MODULE keyword.
- **MODULE(Fullm)** select any module containing a CSECT or ENTRY called FULLM.
- **MODULE(Partm*)** select any module containing a CSECT or ENTRY called PARTM...

ORPHAN select alias members which have no associated main member.

NOORPHAN select main members or alias members which have an associated main member.

OVERLAY select OVERLAY modules.

NOOVERLAY select non-OVERLAY modules.

PAGE select modules marked for loading on a page boundary.

NOPAGE select modules not marked for loading on a page boundary.

REFR select refreshable modules.

NOREFR select non-refreshable modules.

RENT select reentrant modules.

NORENT select non-reentrant modules.

REUS select reusable modules.

NOREUS select non-reusable modules.

RMODE24 select modules with residence mode 24.

NORMODE24 select modules with residence mode ANY.

RMODEANY select modules with residence mode ANY.

NORMODEANY select modules with residence mode 24.

SCTR select scatter-loaded modules.

NOSCTR select no scatter-loaded modules.

SSI(hxdata) select members with matching SSI data. Note that this is implemented as a generic search matching SSI characters from left to right for the number of characters entered.

SSI select members with SSI data.

NOSSI select members without SSI data.

PARTSSI(hxdata) select members with matching SSI data. Note that this is implemented as a pattern search matching SSI digits as a string anywhere in the SSI field of the member.

SYSMOD(Partu) select modules with user IDR data which matches the partial name, **Partu**. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the SYSMOD operand applies only to CSECT's whose names satisfy the MODULE argument.

TEST select modules with the TEST attribute.

NOTEST select modules without the TEST attribute.

TRANS(Partt) select modules with CSECTS identified as having been assembled or compiled by the translator defined by the partial name, **Partt**. **Partt** may be entered as a partial translator IDR name with one to ten characters. Note that if MODULE(...) is also entered, the TRANS operand applies only to CSECT's whose names satisfy the MODULE argument.

TTR(Lttr:Httr) select members whose start address is in the specified TTR range. **Lttr** defaults to 0 and may be entered as a hexadecimal TTR value from 0 through FFFFFF. **Httr** defaults to FFFFFF and may be entered as a hexadecimal TTR value from 0 through FFFFFF.

USERDATA(Partu) select modules with user IDR data which matches the partial name, **Partu**. **Partu** may be entered as partial IDR data with one to eight characters. Note that if MODULE(...) is also entered, the USERDATA operand applies only to CSECT's whose names satisfy the MODULE argument.

USERID(Puid) select members with ISPF statistics and userids matching the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. USERID is an alias of ID.

NOUSERID select members without ISPF statistics. NOUSERID is an alias of NOID.

NOTUSERID(Puid) select members without ISPF statistics or members with ISPF statistics and userids which do not match the partial name, **Puid**. **Puid** may be entered as a partial TSO userid with one to seven characters. NOTUSERID is an alias of NOTID.

VSLKED select modules linked by the MVS OS/VS linkage editor.

NOVSLKED select modules not linked by the MVS OS/VS linkage editor.

ZAP(Partz) select modules with zap IDR data which matches the partial name, **Partz**. **Partz** may be entered as partial zap IDR data with one to eight characters. Note that if MODULE(...) is also entered, the ZAP operand applies only to CSECT's whose names satisfy the MODULE argument.

RESET drop any current members from the member list and add the member group named with this subcommand.

NORESET merge the current member group with the existing member list

LKEDDATE provide linkage edit date info for load members (IDR data must be read).

NOLKEDDATE do not provide that linkage edit date information for load members.

Remarks MEMLIST line commands and their outputs are maintained in the log.

If a STARWARP subcommand is too long for the 48 character input field, you may enter the first 47 or less characters followed by a **+** to indicate more data is required and a continuation panel with more room for operand entry will be provided.

The MEMLIST ISPF table is displayed in response to a MEMLIST subcommand issued in STARWARP line mode or in ISPMODE. Once in the MEMLIST display, you have many options: you may delete a part of the table, find data in the table, add additional members to the table, print a part of the table, store a part of the table in a data set, sort the table in different directions, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

The primary screen for MEMLIST generally shows the most commonly used member attribute information and the alternate screens show some additional data. For source members, the secondary screen adds member TTR information and an ALIASOF field which is filled in for alias members as a result of entering ALIASCHK (alias check or option 55). The primary screen fields are as follows:

CMD command. This is where a line command would be entered.

NAME member name. The eight character member name.

DATA/MSG data field. Up to eight bytes of operand information can be entered here. It is used as a command feedback field -- any information in this field which is preceded by an asterisk (*) is ignored when a line command is entered.

VER.MOD version and modification level. These numbers can range from 0 through 99 and they reflect the level of the member.

CREATED creation date. The format is YY/MM/DD.

LAST modification date. The format is YY/MM/DD.

MODIFIED modification time. The format is HH:MM.

SIZE size of member in lines. Member sizes up to 65,555 can be displayed.

INIT initial size of member in lines. Member sizes up to 65,555 can be displayed.

ID userid. The userid of the last person to modify this member.

The second source member screen reformats the data and adds the following data for each member:

ALIASOF alias information field. For aliases, this field will contain *ALIAS until a ALIASCHK (option 55) is performed. After that point, it will contain the name of the corresponding main member if it exists or *ORPHAN otherwise.

TTR TTR address. The address of the first record for this member relative to the start of the data set.

Load members have four different screens defined. The first screen contains the linkage edit date and most information required to manipulate load members. The second screen adds more detailed information; the third screen combines the first two screens and the fourth screen may be customized and contains Change Man data.

The first screen displays the following data:

CMD command. This is where a line command would be entered.

NAME member name. The eight character member name.

DATA/MSG data field. Up to eight bytes of operand information can be entered here. It is used as a command feedback field -- any information in this field which is preceded by an asterisk (*) is ignored when a line command is entered.

ALIASOF alias information field. For aliases, this field will contain *ALIAS until a ALIASCHK (option 55) is performed. After that point, it will contain the name of the corresponding main member if it exists or one of the following:

***ORPHAN** orphan. No corresponding main member is in the data set.

=BYTTR orphan. However, a main member does match by TTR address but it does not have the same name as noted in the orphan's directory entry.

=BYNAME orphan. However, the main member noted in the directory entry of the orphan member exists but it does not have the same TTR address.

LEN/LKED length/LKED date. If LKEDDATE is in effect, the IDR data for each member is read and this field will contain the linkage edit date in YY/MM/DD format. Otherwise, the module length will be displayed using six hexadecimal characters.

ATTRIBUTES linkage attributes. This is actually four columns which will contain linkage attributes as follows:

DC	Downward compatible attribute.
LOAD	Load only attribute.
NOED	NOEDIT attribute.
NOEX	NOT Executable attribute.
OVLY	Overlay attribute.
REFR	Refreshable attribute.
RENT	Reentrant attribute.
REUS	Reusable attribute.
SCTR	Scatter attribute.
ERRL	Load only attribute conflict with scatter attribute.
ERRO	Overlay attribute conflict with reentrant, reusable, load only or scatter attribute.
ERRS	Scatter attribute conflict with reentrant, overlay or load only.
ERR1	Refreshable attribute conflict with not reusable attribute.

APF This field will contain AC=1 if the module is marked as authorized; if the field is incorrectly formatted it will contain ERR; otherwise, this field will be blank.

MODE This field contains an indication of the residence mode of the module; if AMODE and RMODE have not been set this field will be blank, otherwise, this field will contain one of the following:

RANY	RMODE of ANY and AMODE 31.
AANY	RMODE of 24 and AMODE ANY.
A31	RMODE of 24 and AMODE 31.
A24	RMODE of 24 and AMODE 24.
ERR	RMODE of ANY; AMODE ANY or AMODE of 24. This is invalid.

MAIN This field is for alias entries. It contains the name of the corresponding main member as noted in the directory entry. This information can be checked for correctness by ALIASCHK (option 55).

The second load module screen adds the following fields:

TTR TTR address. The address of the first record for this member relative to the start of the data set.

MATCH Matching member by TTR. This field is for alias entries; it is filled in by ALIASCHK (option 55) to show the name of the matching main member according to the TTR address. If the ALIASOF field has an unusual value (such as =BYTTR or =BYNAME), this field and the MAIN field will provide names of the affected members.

LENGTH Module length. Hexadecimal length of the module.

LEN-KB Module length in 1024 or K-byte units.

ENTRY Entry point address in hexadecimal.

SSI SSI data in hexadecimal.

MEMLIST tables can be saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE and MERGE commands.

Saved MEMLIST tables are given members names of the form **PDSML0na** if the name is one or two numeric digits; otherwise, a name of the form **@#name** is used where name is a one to six character alphameric table identification name.

The following primary commands are supported directly for the MEMLIST function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

***** merges current member group members into the MEMLIST table.

ALIASCHK Alias check. Checks aliases and adds all associated members to the MEMLIST display.

ALL MEMLIST all. Adds all members in the data set to the MEMLIST display.

ALT[ERNAT] displays an alternate view of the MEMLIST table

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

BAT[CHJCL] builds batch JCL which will include all STARWARP statements in the LOG table.

DUA[L] displays a double line view of the MEMLIST table.

EDITT[BL] (or **ET[BL]**) enters an edit session on MEMLIST table data.

EQ[UATE] SUBLIST =. Resets the current member group to the members in the current MEMLIST.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
  [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
  [ANY/MEMBER/DATE/LENGTH/ENTRY/MAIN/
   /MATCH/SSI/TTR/ALIASOF]          (for load)
  [ANY/MEMBER/VERMOD/CREATED/MODIFIED/
   /SIZE/INIT/ID/TTR/ALIASOF]       (for source)
```

LE[FT] rotates through alternate views of the MEMLIST table. PF keys 10 and 22 are normally set to LEFT.

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax:
```
LOCATE pmember/ttr/puserid/pdate/size/vermod
       /plinkdate/hexlength/apf/attrib/pssi
```

MER[GE] add members to the MEMLIST table from a saved MEMLIST table.
Syntax: `MERGE name [RESET] [NOSTATS]`
(where name is one to six alphameric characters)

MON[TH] merges members updated or created this month into the MEMLIST.

NOR[MAL] displays the default view of the MEMLIST table.

O[PTIONS] provides primary command selection for the MEMLIST function and operand syntax assistance.

OUT[PUT] outputs the MEMLIST table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the MEMLIST table based on a string match.
Syntax:
```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
       [MEMBER/VERMOD/CREATED/MODIFIED/SIZE/INIT/
        ID/TTR/ALIASOF/DATA/MSGS] (source members)
       [MEMBER/DATE/LENGTH/ENTRY/MAIN/MATCH/SSI/
        TTR/ALIASOF/DATA/MSGS] (load members)
```

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the MEMLIST table. PF keys 11 and 23 are normally set to RIGHT.

SAVE creates a permanent table for use in a different STARWARP session.
Syntax: `SAVE [name] [REPLACE/NOREPL]`
(where name is one to six alphameric characters)

S[ELECT] normally BROWSEs load members and EDITs source members. The value used for the command is set in SETSEL.
Syntax: `SELECT member`

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.
Syntax: SORT
```
[MEMBER/TTR/DATE/SIZE/APF/MODE/ATTR/TTR]  (load)
[MEMBER/TTR/VERMOD/CREATED/MODIF/SIZE/ID] (source)
[ASCEND/DESCEND]
```

TAG forms a member group containing only members marked with *TAG* in the DATA/MSG field.

TOD[AY] merges members updated or created today into the MEMLIST.

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation. The MEMLIST UT panel also allows dynamic primary commands with which you specify command names and their corresponding actions.

WE[EK] merges members updated or created this week into the MEMLIST.

X clears the MEMLIST table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the MEMLIST table; this is equivalent to **X ALL**

In addition to the above, STARWARP subcommands and ISPF commands may be entered on the command line. Also, if you want a STARWARP subcommand to apply to all members currently in the MEMLIST, use = as the member group name for the subcommand.

By default, line commands which produce output display it in an edit session and this output is also recorded in the log. To have this output to be placed only in the log, check the setting of "Automatic EDITLOG" in SETALL for MEMLIST. The following line commands are supported in the MEMLIST function:

+ provide an extension panel for command entry.
= repeat the previous line command.
ATTR provide a preinitialized entry assist panel to update member attributes.
CSEC invoke the CSECTS function for a load member.
DEC decrypt the member into another data set.
ENC encrypt the member into another data set.
INFO display attributes of the member using the ATTRIB subcommand.
K kill and clear all following line commands.
LOG copy the line into the log.

M provide line command selection and entry assistance.

ML update statistics and attributes for this member.

O provide line command selection and operand syntax assistance.

PAN display this member as a panel using ISPF services.

S select line command (normally BROWSE for load and EDIT for source; see SETSEL).

SWAP rename swap this member with the member named in the DATA/MSG field.

TAG mark this member line with *TAG* in the DATA/MSG field.

UT select the extended user line command panel.

W add the data set name and the member name to the WORKPAD.

X drop this member from the MEMLIST table.

XMIT transmit this member to another user.

In addition to the above line commands, most STARWARP subcommands which define a "memgroup" may be used as line commands; this set of subcommands includes BROWSE, DCF, DELINK, DIRENTRY, DISASM, EDIT, FIND, FSE, HISTORY, LIST, MAP, PGMDOC, PRINT, READOBJ, REPLACE, REVIEW, SUBMIT, TSOEDIT, TSOLIST, VERIFY, VPRINT and XREF.

Also, the ALIAS and COMPARE subcommands may be used as line commands but they each require another member name to be entered as an operand in the DATA/MSG field.

Note that each subcommand name must be abbreviated to a maximum of four characters to fit into the CMD field. Also, note that several of the above subcommands are available only on source data sets (DCF, EDIT, FSE, READOBJ, SUBMIT and TSOEDIT) and that others are available only on load data sets (DELINK, DISASM, HISTORY, MAP, PGMDOC and XREF).

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the MEMLIST function:

== = command, repeat the previous line command for the range of lines.

AA ATTRIB subcommand, display attributes.

BB BROWSE subcommand, enter ISPF browse.

DD DIRENTRY subcommand, display the directory entry.

DDEL DEL command, delete each member.

EE EDIT subcommand, enter ISPF edit (source only).

HH HISTORY subcommand, display module history data (load only).

LL LIST command, list each member.

MM MAP subcommand, display module map (load only).

OO O command, provide line command assistance for each line.

PP PRINT subcommand, produce a hard-copy.

SS S command, select a member (normally edit or browse).

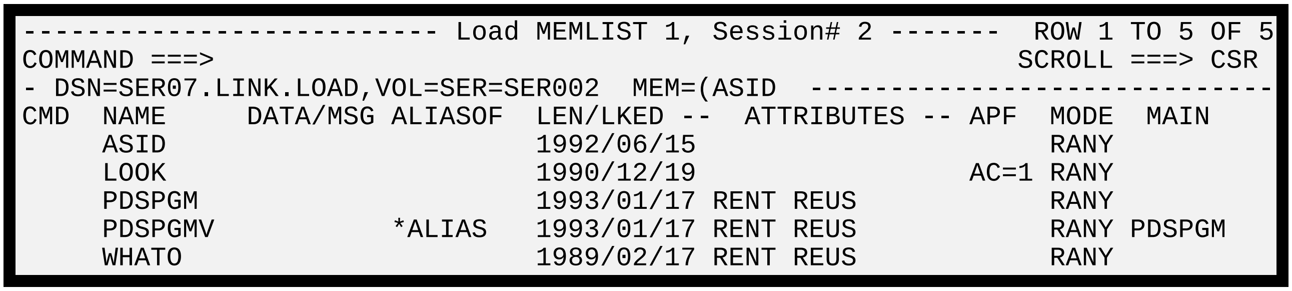
TT TAG command, mark each member line with *TAG* in the DATA/MSG field.

VV VERIFY subcommand, validate each member.

XX X command, drop the range of table lines.

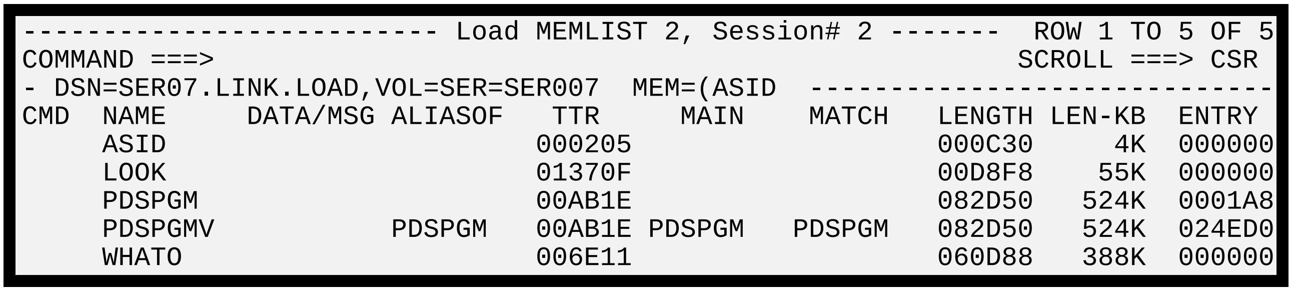
MEMLIST : RMODEANY was entered to produce the following figure:

MEMLIST Function



```
------------------------- Load MEMLIST 1, Session# 2 -------  ROW 1 TO 5 OF 5
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER002  MEM=(ASID  -----------------------------
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED --  ATTRIBUTES -- APF  MODE  MAIN
     ASID                       1992/06/15                      RANY
     LOOK                       1990/12/19                 AC=1 RANY
     PDSPGM                     1993/01/17 RENT REUS            RANY
     PDSPGMV           *ALIAS   1993/01/17 RENT REUS            RANY PDSPGM
     WHATO                      1989/02/17 RENT REUS            RANY
```

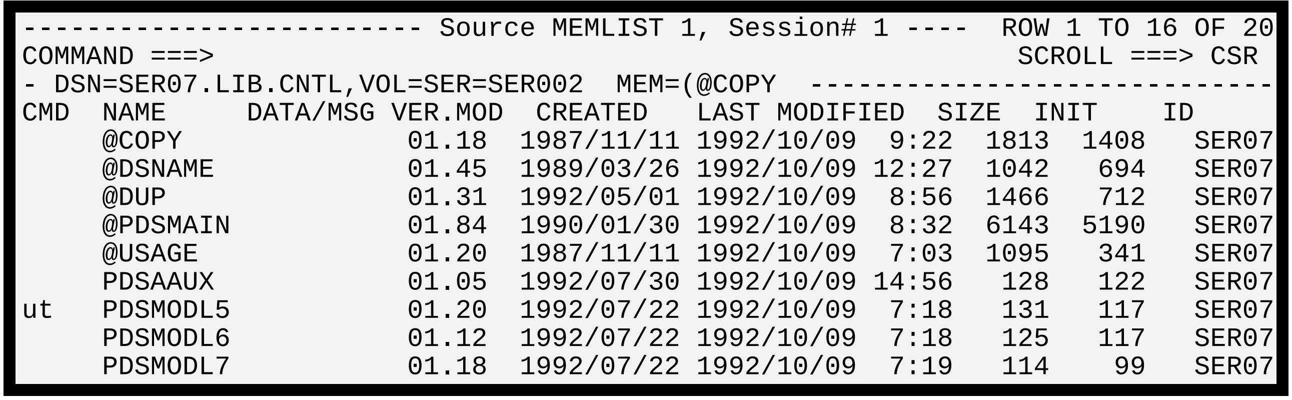
Figure 111. Sample MEMLIST Table (load)



```
------------------------- Load MEMLIST 2, Session# 2 -------  ROW 1 TO 5 OF 5
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=(ASID  -----------------------------
CMD  NAME     DATA/MSG ALIASOF   TTR     MAIN    MATCH   LENGTH LEN-KB  ENTRY
     ASID                       000205                   000C30     4K  000000
     LOOK                       01370F                   00D8F8    55K  000000
     PDSPGM                     00AB1E                   082D50   524K  0001A8
     PDSPGMV           PDSPGM   00AB1E PDSPGM   PDSPGM   082D50   524K  024ED0
     WHATO                      006E11                   060D88   388K  000000
```

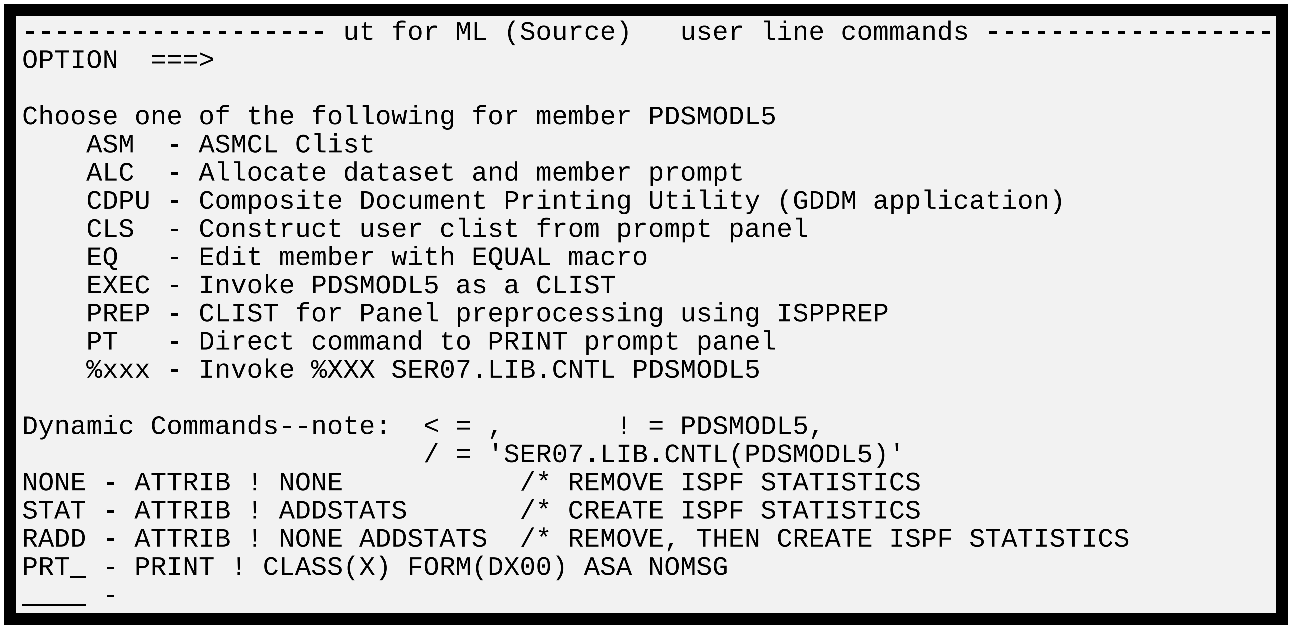
Figure 112. Sample MEMLIST Table (load-RIGHT view)

TODAY was entered to produce the following figure:



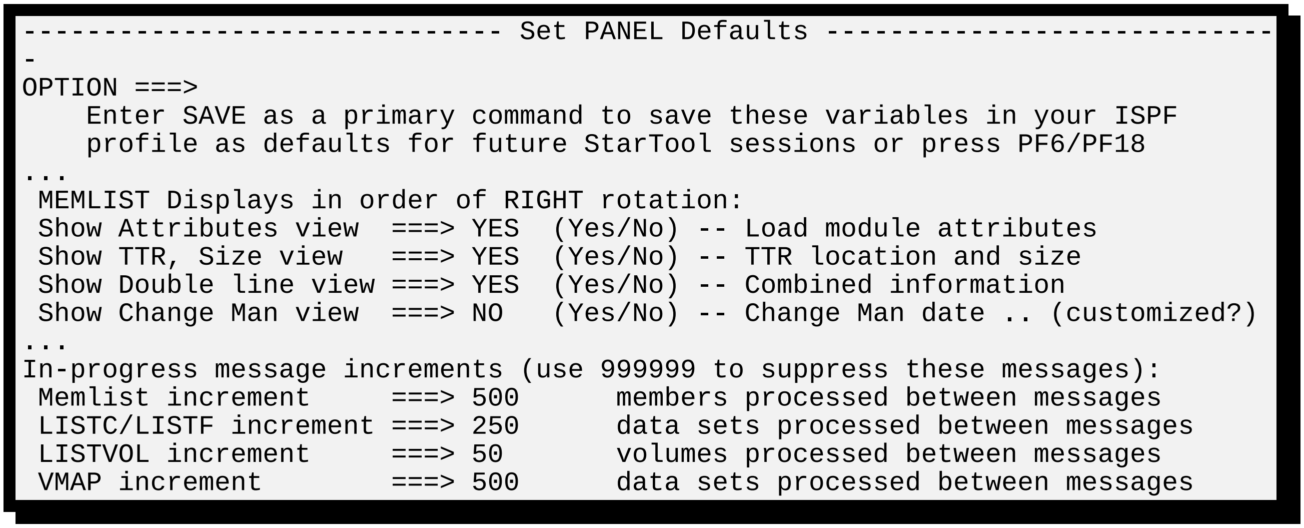
```
------------------------ Source MEMLIST 1, Session# 1 ----  ROW 1 TO 16 OF 20
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER002  MEM=(@COPY  -----------------------------
CMD  NAME     DATA/MSG VER.MOD  CREATED   LAST MODIFIED  SIZE  INIT    ID
     @COPY              01.18  1987/11/11 1992/10/09  9:22  1813  1408   SER07
     @DSNAME            01.45  1989/03/26 1992/10/09 12:27  1042   694   SER07
     @DUP               01.31  1992/05/01 1992/10/09  8:56  1466   712   SER07
     @PDSMAIN           01.84  1990/01/30 1992/10/09  8:32  6143  5190   SER07
     @USAGE             01.20  1987/11/11 1992/10/09  7:03  1095   341   SER07
     PDSAAUX            01.05  1992/07/30 1992/10/09 14:56   128   122   SER07
ut   PDSMODL5           01.20  1992/07/22 1992/10/09  7:18   131   117   SER07
     PDSMODL6           01.12  1992/07/22 1992/10/09  7:18   125   117   SER07
     PDSMODL7           01.18  1992/07/22 1992/10/09  7:19   114    99   SER07
```

Figure 113. Sample MEMLIST Table (source)



```
------------------- ut for ML (Source)   user line commands ------------------
OPTION  ===>

Choose one of the following for member PDSMODL5
    ASM  - ASMCL Clist
    ALC  - Allocate dataset and member prompt
    CDPU - Composite Document Printing Utility (GDDM application)
    CLS  - Construct user clist from prompt panel
    EQ   - Edit member with EQUAL macro
    EXEC - Invoke PDSMODL5 as a CLIST
    PREP - CLIST for Panel preprocessing using ISPPREP
    PT   - Direct command to PRINT prompt panel
    %xxx - Invoke %XXX SER07.LIB.CNTL PDSMODL5

Dynamic Commands--note:  < = ,       ! = PDSMODL5,
                         / = 'SER07.LIB.CNTL(PDSMODL5)'
NONE - ATTRIB ! NONE           /* REMOVE ISPF STATISTICS
STAT - ATTRIB ! ADDSTATS       /* CREATE ISPF STATISTICS
RADD - ATTRIB ! NONE ADDSTATS  /* REMOVE, THEN CREATE ISPF STATISTICS
PRT_ - PRINT ! CLASS(X) FORM(DX00) ASA NOMSG
____ -
```

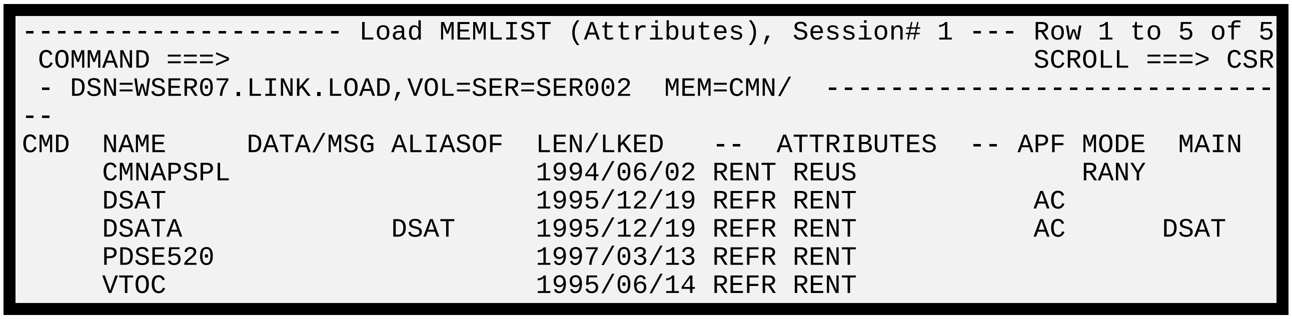
Figure 114. Sample MEMLIST user line command



```
------------------------------ Set PANEL Defaults ----------------------------
-
OPTION ===>
    Enter SAVE as a primary command to save these variables in your ISPF
    profile as defaults for future StarTool sessions or press PF6/PF18
...
 MEMLIST Displays in order of RIGHT rotation:
 Show Attributes view  ===> YES  (Yes/No) -- Load module attributes
 Show TTR, Size view   ===> YES  (Yes/No) -- TTR location and size
 Show Double line view ===> YES  (Yes/No) -- Combined information
 Show Change Man view  ===> NO   (Yes/No) -- Change Man date .. (customized?)
...
In-progress message increments (use 999999 to suppress these messages):
 Memlist increment     ===> 500      members processed between messages
 LISTC/LISTF increment ===> 250      data sets processed between messages
 LISTVOL increment     ===> 50       volumes processed between messages
 VMAP increment        ===> 500      data sets processed between messages
```

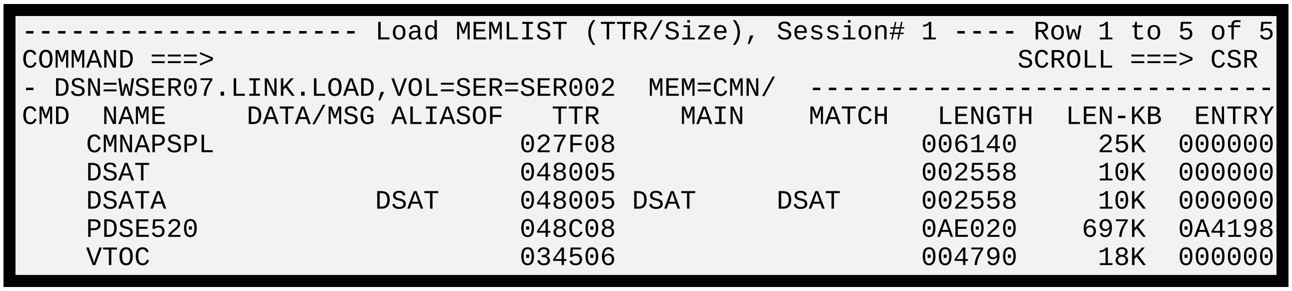
Figure 115. Partial SETPANEL panel with MEMLIST load defaults

MEMLIST Function



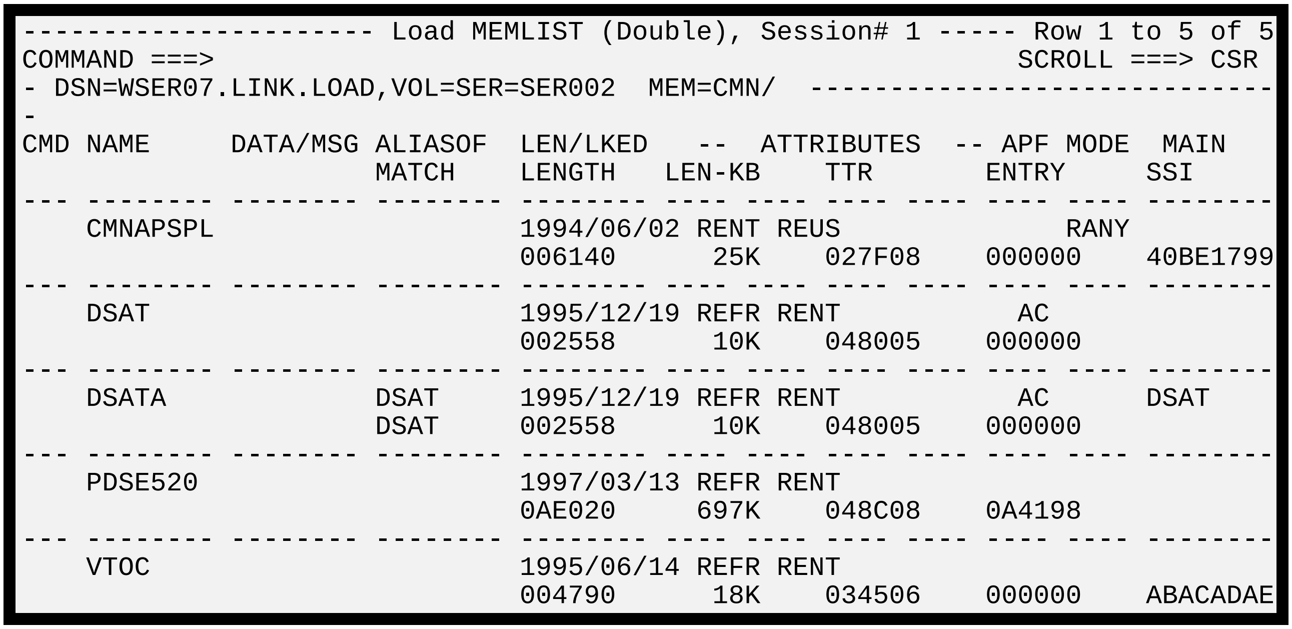
```
-------------------- Load MEMLIST (Attributes), Session# 1 --- Row 1 to 5 of 5
 COMMAND ===>                                                   SCROLL ===> CSR
 - DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/  -----------------------------
--
CMD  NAME     DATA/MSG ALIASOF  LEN/LKED   --  ATTRIBUTES  -- APF MODE  MAIN
     CMNAPSPL                   1994/06/02 RENT REUS                RANY
     DSAT                       1995/12/19 REFR RENT           AC
     DSATA             DSAT     1995/12/19 REFR RENT           AC      DSAT
     PDSE520                    1997/03/13 REFR RENT
     VTOC                       1995/06/14 REFR RENT
```

Figure 116. Sample MEMLIST (Attributes view--On by default)



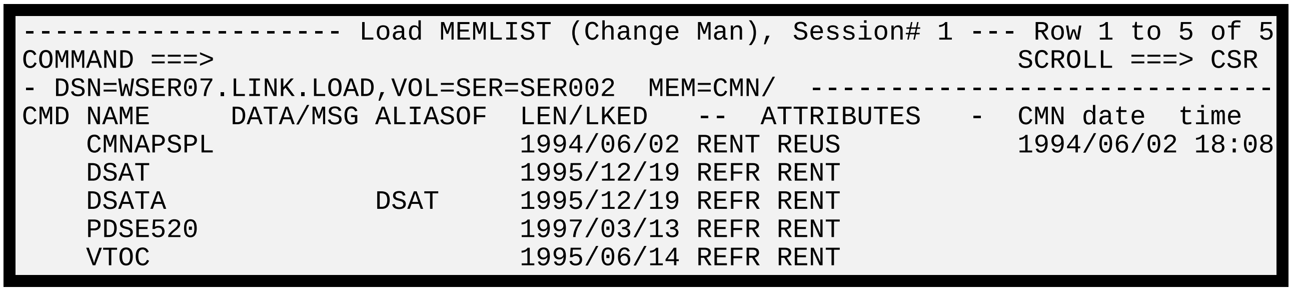
```
--------------------- Load MEMLIST (TTR/Size), Session# 1 ---- Row 1 to 5 of 5
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/  ------------------------------
CMD  NAME     DATA/MSG ALIASOF   TTR     MAIN    MATCH   LENGTH  LEN-KB  ENTRY
    CMNAPSPL                   027F08                   006140     25K  000000
    DSAT                       048005                   002558     10K  000000
    DSATA             DSAT     048005 DSAT     DSAT     002558     10K  000000
    PDSE520                    048C08                   0AE020    697K  0A4198
    VTOC                       034506                   004790     18K  000000
```

Figure 117. Sample MEMLIST (TTR/Size view--On by default)



```
---------------------- Load MEMLIST (Double), Session# 1 ----- Row 1 to 5 of 5
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/  ------------------------------
-
CMD NAME     DATA/MSG ALIASOF  LEN/LKED   --  ATTRIBUTES  -- APF MODE  MAIN
                      MATCH    LENGTH   LEN-KB    TTR       ENTRY     SSI
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
    CMNAPSPL                   1994/06/02 RENT REUS              RANY
                               006140      25K    027F08    000000    40BE1799
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
    DSAT                       1995/12/19 REFR RENT           AC
                               002558      10K    048005    000000
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
    DSATA             DSAT     1995/12/19 REFR RENT           AC      DSAT
                      DSAT     002558      10K    048005    000000
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
    PDSE520                    1997/03/13 REFR RENT
                               0AE020     697K    048C08    0A4198
--- -------- -------- -------- -------- ---- ---- ---- ---- ---- ---- --------
    VTOC                       1995/06/14 REFR RENT
                               004790      18K    034506    000000    ABACADAE
```

Figure 118. Sample MEMLIST (Double view--On by default)



```
-------------------- Load MEMLIST (Change Man), Session# 1 --- Row 1 to 5 of 5
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.LINK.LOAD,VOL=SER=SER002  MEM=CMN/  -----------------------------
CMD NAME     DATA/MSG ALIASOF  LEN/LKED   --  ATTRIBUTES   -  CMN date  time
    CMNAPSPL                   1994/06/02 RENT REUS           1994/06/02 18:08
    DSAT                       1995/12/19 REFR RENT
    DSATA             DSAT     1995/12/19 REFR RENT
    PDSE520                    1997/03/13 REFR RENT
    VTOC                       1995/06/14 REFR RENT
```

Figure 119. Sample MEMLIST (Change Man view--Off by default)

NUCMAP Function

Purpose The NUCMAP command displays the current system nucleus.

Example NUCMAP

Syntax

```
NUCMAP
```

Aliases NUC, NUCM, NUCMA, NUCMAP

Defaults none

Required none

Operands none

Remarks NUCMAP provides a way to look at the current nucleus by memory address. The CSECTS table is used.

The CSECTS ISPF table is displayed in response to a NUCMAP command. When you are in a CSECTS/NUCMAP display, you have many options: you may delete a part of the table, sort the table in different directions, find data in the table, print a part of the table or store a part of the table in a data set. For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

The CSECTS/NUCMAP table displays the names of all CSECTs and ENTRYs for the NUCLEUS. The screen fields for each line are as follows:

CMD command. This is where a line command would be entered.

TYPE type of entry. Any of the following may appear.

- **CSECT** CSECT entry.
- **ENTRY** ENTRY within a CSECT.

DATA/MSG data field. This is a command feedback field.

SECTION CSECT name.

ENTRY ENTRY name.

ADDR Address of this symbol (this is for the CSECTS function only).

LENG Hexadecimal length of this CSECT.

MD/SG Mode or segment. An AMODE/RMODE indication is provided as follows:

- **RANY** RMODE of ANY and AMODE 31.
- **AANY** RMODE of 24 and AMODE ANY.
- **A31** RMODE of 24 and AMODE 31.
- **A24** RMODE of 24 and AMODE 24.

MEM ADDR Memory address in hexadecimal.

The following primary commands are supported directly for the CSECTS/NUCMAP function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

EDITT[BL] (or **ET[BL]**) enters an edit session on CSECT/NUCMAP table data.

EXPR[ESS] executes all entered line commands without pauses between individual commands.

F finds a string and positions the display start location.
Syntax:

```
F anystring [ASIS]
   [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
   [ANY/ADDRESS/NAME/ENTRY/SEGMENT/MODE]
```

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax: `LOCATE csectname/address`

O[PTIONS] provides primary command selection for the CSECTS/NUCMAP function and operand syntax assistance.

OUT[PUT] outputs the CSECTS/NUCMAP table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the CSECTS/NUCMAP table based on a string match.
Syntax:

```
REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
          [ANY/ADDRESS/NAME/ENTRY/SEGMENT/MODE]
```

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.
Syntax:

```
SORT [ADDRESS/NAME]
     [ASCEND/DESCEND]
```

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.

X clears the CSECTS/NUCMAP table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

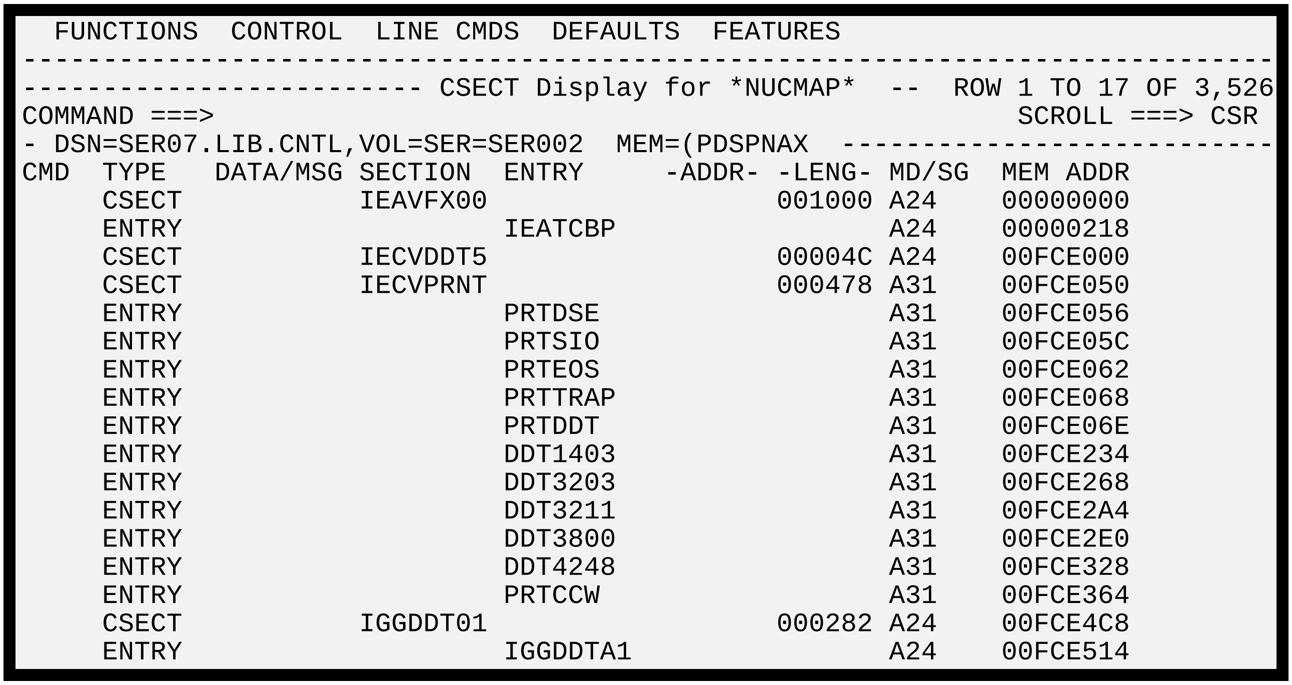
XA[LL] clears the CSECTS/NUCMAP table; this is equivalent to **X ALL**

The following line commands are supported in the CSECTS/NUCMAP function:

+ provide an extension panel for command entry.
= repeat the previous line command.
K kill and clear all following line commands.
LOG copy the line into the log.
O provide line command selection and operand syntax assistance.
TAG mark this table entry with *TAG* in the DATA/MSG field.
X drop the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the CSECTS function:

== = command, repeat the previous line command for the range of lines.
OO O command, provide line command assistance for each line.
XX X command, drop the range of table lines.



```
  FUNCTIONS  CONTROL  LINE CMDS  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- CSECT Display for *NUCMAP*  --  ROW 1 TO 17 OF 3,526
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER002  MEM=(PDSPNAX  ---------------------------
CMD  TYPE   DATA/MSG SECTION  ENTRY     -ADDR- -LENG- MD/SG  MEM ADDR
     CSECT           IEAVFX00                 001000 A24    00000000
     ENTRY                    IEATCBP                A24    00000218
     CSECT           IECVDDT5                 00004C A24    00FCE000
     CSECT           IECVPRNT                 000478 A31    00FCE050
     ENTRY                    PRTDSE                 A31    00FCE056
     ENTRY                    PRTSIO                 A31    00FCE05C
     ENTRY                    PRTEOS                 A31    00FCE062
     ENTRY                    PRTTRAP                A31    00FCE068
     ENTRY                    PRTDDT                 A31    00FCE06E
     ENTRY                    DDT1403                A31    00FCE234
     ENTRY                    DDT3203                A31    00FCE268
     ENTRY                    DDT3211                A31    00FCE2A4
     ENTRY                    DDT3800                A31    00FCE2E0
     ENTRY                    DDT4248                A31    00FCE328
     ENTRY                    PRTCCW                 A31    00FCE364
     CSECT           IGGDDT01                 000282 A24    00FCE4C8
     ENTRY                    IGGDDTA1               A24    00FCE514
```

Figure 120. Sample NUCMAP Table

OPTIONS Subcommand

Purpose The OPTIONS subcommand displays the subcommands (or options) supported by STARWARP. Note that this display varies when a non-partitioned data set is being processed.

Example OPTIONS

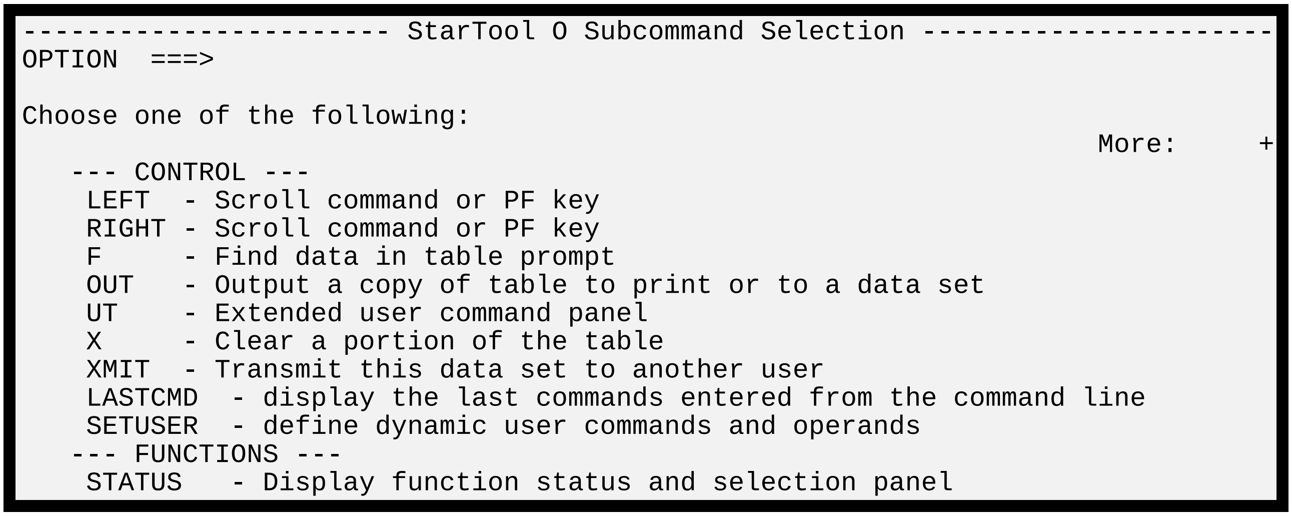
Syntax

```
OPTIONS
```

Aliases O, OP, OPT, OPTI, OPTIO, OPTION, OPTIONS

Operands (no operands are supported for the OPTIONS subcommand).

Remarks When in ISPMODE or MEMLIST, if the O alias of OPTIONS is entered after **COMMAND ===>**, a panel of options is presented to assist in entering STARWARP subcommands.
When in any table function, if the o alias of OPTIONS is entered as a line command, a panel of line command options is presented to assist in entering subcommands.



```
----------------------- StarTool O Subcommand Selection ----------------------
OPTION  ===>

Choose one of the following:
                                                                 More:     +
   --- CONTROL ---
    LEFT  - Scroll command or PF key
    RIGHT - Scroll command or PF key
    F     - Find data in table prompt
    OUT   - Output a copy of table to print or to a data set
    UT    - Extended user command panel
    X     - Clear a portion of the table
    XMIT  - Transmit this data set to another user
    LASTCMD  - display the last commands entered from the command line
    SETUSER  - define dynamic user commands and operands
   --- FUNCTIONS ---
    STATUS   - Display function status and selection panel
```

Figure 121. Sample OPTIONS Subcommand

PATTERN Subcommand

Purpose The PATTERN subcommand lists member names from the data set directory. Only those member names which contain the PATTERN name segments specified are displayed. Note that the default member group is not affected by the DISPLAY or PATTERN subcommands.

One or two member name segments may be entered as patterns. If two segments are entered, both segments must occur in a member name for it to be displayed. For example, with the pattern segments xx and 1x, members named M1X4XX, M1XX and XX1X would be displayed by this subcommand while members named X1X or MXX13X would not be displayed.

Example PATTERN iea/sys

Syntax

```
PATTERN   [segment1 [segment2] ]
```

Aliases P, PA, PAT, PATT, PATTE, PATTER, PATTERN

Defaults previously entered segment1 and segment2 operands

Required none

Operands

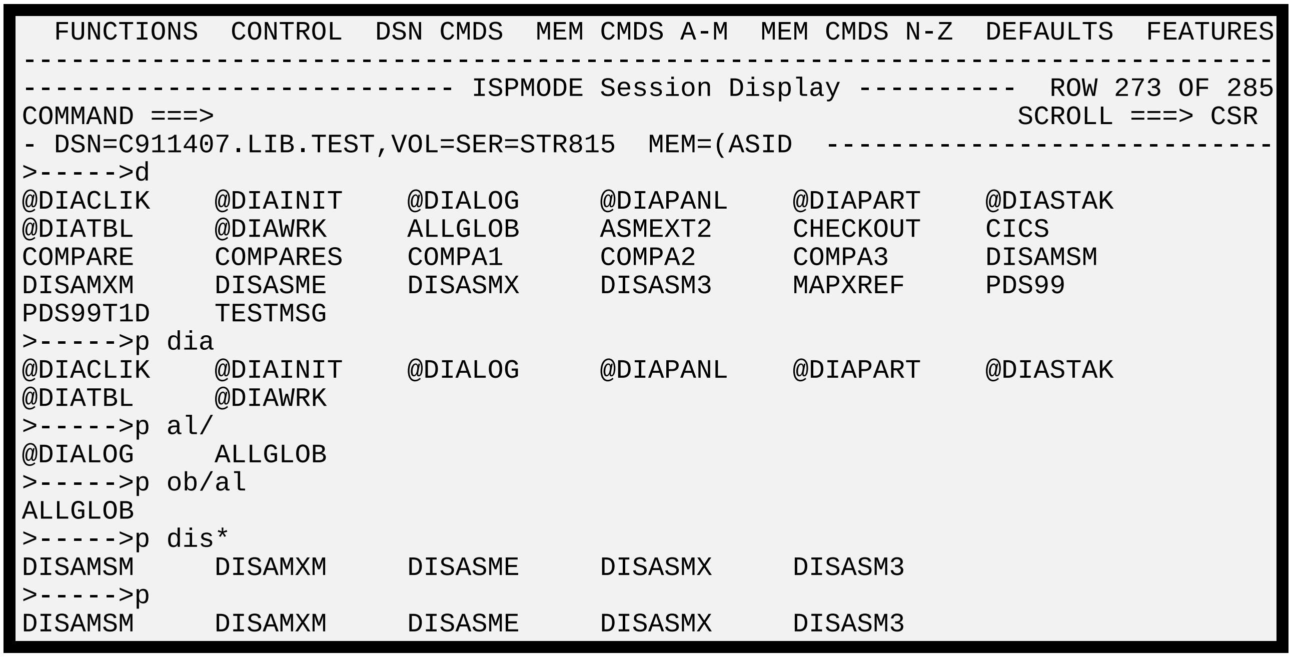
segment1 specifies a name segment which must be found in a member name for the member name to be displayed.

Note: a group name descriptor such as **part1:part2**, **seg1/seg2** or **part1*seg1** may be entered for segment1, in which case, the output will consist of the member names which would be displayed by the MEMBERS subcommand.

segment2 specifies a second name segment which must also be found in a member name for the member name to be displayed.

Remarks Neither PATTERN nor DISPLAY affect the current member group. If PATTERN is entered without operands, the previously entered operand(s) are used. This differs from DISPLAY; a DISPLAY with no operands will show all members in the data set.

Note that **-A** is added following a displayed member name if the member name is an alias. The member name is listed in a combined hexadecimal/character format if it does not contain upper case alphameric characters (including @, $ and #) or if the first character is numeric. Unprintable characters (for 3270-type devices) in a member name are displayed as periods.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 273 OF 285
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.TEST,VOL=SER=STR815  MEM=(ASID  ----------------------------
>----->d
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK     ALLGLOB     ASMEXT2     CHECKOUT    CICS
COMPARE     COMPARES    COMPA1      COMPA2      COMPA3      DISAMSM
DISAMXM     DISASME     DISASMX     DISASM3     MAPXREF     PDS99
PDS99T1D    TESTMSG
>----->p dia
@DIACLIK    @DIAINIT    @DIALOG     @DIAPANL    @DIAPART    @DIASTAK
@DIATBL     @DIAWRK
>----->p al/
@DIALOG     ALLGLOB
>----->p ob/al
ALLGLOB
>----->p dis*
DISAMSM     DISAMXM     DISASME     DISASMX     DISASM3
>----->p
DISAMSM     DISAMXM     DISASME     DISASMX     DISASM3
```

Figure 122. Sample PATTERN Subcommand

PBROWSE Function

Purpose The SUPEREDIT option includes the PEDIT and PBROWSE subcommands which can be invoked transparently with the EDIT and BROWSE subcommands. PBROWSE (for parallel browse) operates under control of STARWARP; however, PEDIT (for parallel edit) is only available with STARTOOL.

PBROWSE supports sequential, direct, PDS, PDSE and all VSAM data types including PATH's. Also, you can perform VSAM positioning. PBROWSE is very similar to ISPF browse, except that additional data types are supported. PB (for PBROWSE) may be used as a line command in MEMLIST; it also supports deleted members.

Example PBROWSE

Syntax

```
PBROWSE   member
       * [FROMKEY(key)/FROMADDRESS(add)/FROMNUMBER(num)  ]
       * [TOKEY(key)/TOADDRESS(add)/TONUMBER(num)        ]
*NOTE: Lines with an asterisk are supported for VSAM data sets only
```

Aliases PB, PBR, PBRO, PBROW, PBROWS, PBROWSE

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults none

Required member

Operands

member identifies the member to be browsed.

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x’hexkey’**; access begins at the first record whose key matches (or is greater than) the portion of the key specified. This parameter may be used with TOKEY; it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.

This parameter may be used with TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified.

This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.

This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

Remarks PBROWSE is called a parallel function because you can suspend a browse session to process other data sets using GO sessions. On returning control to the browse session, you can resume where you left off as with other STARWARP parallel activities.

It is suggested that a LOG command be used to suspend PBROWSE; you can then transit to other STARWARP processes; however, you can actually use any function name or subcommand which does not conflict with the command names supported for PBROWSE.

The following primary commands are supported directly in the PBROWSE function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 253.

CO[LS] displays a columns line on the first line of the data area. The columns line will remain at the top of the data display; it is useful in identifying columns to be used with the FIND command.
Syntax: `COLS [ON/OFF]`

F[IND] finds a string and positions the display start location.

```
Syntax: FIND anystring [nn mm] [ASIS]
             [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
```

(where nn and mm are column numbers)

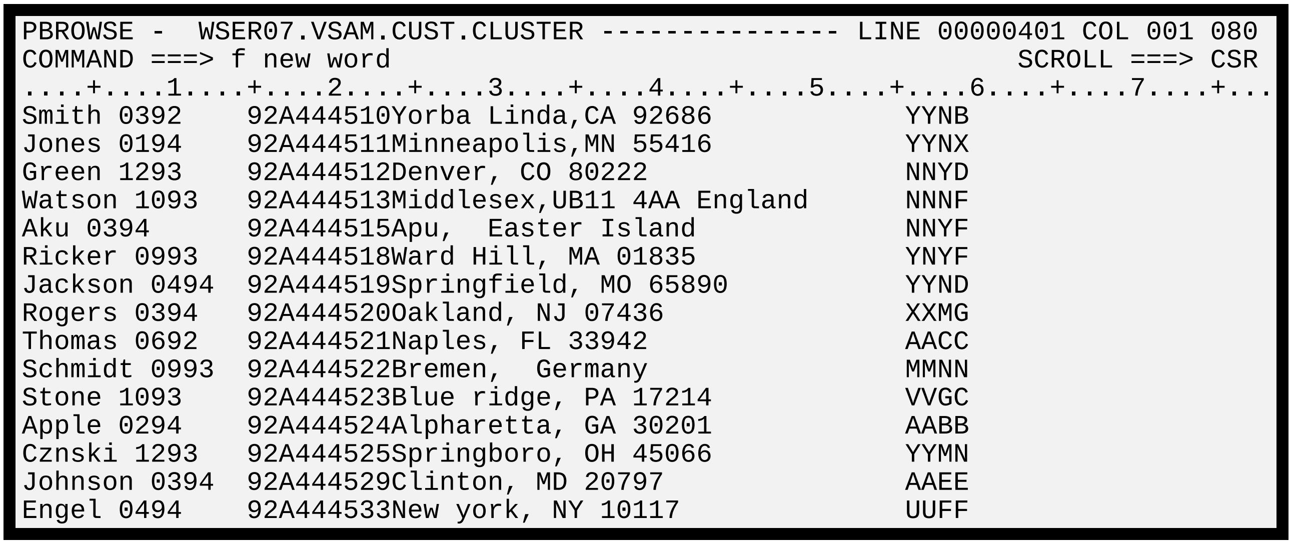
HEX sets HEX display mode on or off.
Syntax: `HEX [ON/OFF]`

L[OCATE] positions directly to the specified record number.
Syntax: `LOCATE recnumber`

PBROWSE Function

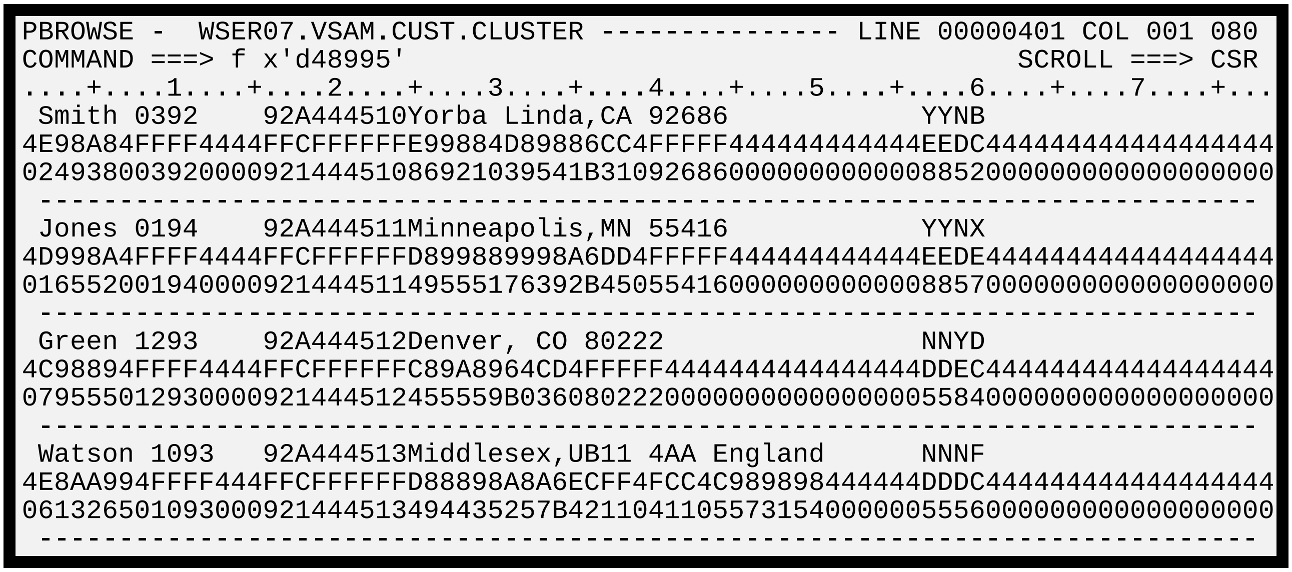
PAD changes the padding character and specifies the number of pad characters desired.
Syntax: `PAD character/'character'/x'hex' [column]`
Example: PAD a 45

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.



```
PBROWSE -  WSER07.VSAM.CUST.CLUSTER --------------- LINE 00000401 COL 001 080
COMMAND ===> f new word                                       SCROLL ===> CSR
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
Smith 0392    92A444510Yorba Linda,CA 92686            YYNB
Jones 0194    92A444511Minneapolis,MN 55416            YYNX
Green 1293    92A444512Denver, CO 80222                NNYD
Watson 1093   92A444513Middlesex,UB11 4AA England      NNNF
Aku 0394      92A444515Apu,  Easter Island             NNYF
Ricker 0993   92A444518Ward Hill, MA 01835             YNYF
Jackson 0494  92A444519Springfield, MO 65890           YYND
Rogers 0394   92A444520Oakland, NJ 07436               XXMG
Thomas 0692   92A444521Naples, FL 33942                AACC
Schmidt 0993  92A444522Bremen,  Germany                MMNN
Stone 1093    92A444523Blue ridge, PA 17214            VVGC
Apple 0294    92A444524Alpharetta, GA 30201            AABB
Cznski 1293   92A444525Springboro, OH 45066            YYMN
Johnson 0394  92A444529Clinton, MD 20797               AAEE
Engel 0494    92A444533New york, NY 10117              UUFF
```

Figure 123. Sample PBROWSE Subcommand



```
PBROWSE -  WSER07.VSAM.CUST.CLUSTER --------------- LINE 00000401 COL 001 080
COMMAND ===> f x'd48995'                                      SCROLL ===> CSR
....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
 Smith 0392    92A444510Yorba Linda,CA 92686            YYNB
4E98A84FFFF4444FFCFFFFFFE99884D89886CC4FFFFF444444444444EEDC444444444444444444
024938003920000921444510869210395 41B3109268600000000000008852000000000000000000
 ----------------------------------------------------------------------------
 Jones 0194    92A444511Minneapolis,MN 55416            YYNX
4D998A4FFFF4444FFCFFFFFFD899889998A6DD4FFFFF444444444444EEDE444444444444444444
016552001940000921444511495551763 92B4505541600000000000008857000000000000000000
 ----------------------------------------------------------------------------
 Green 1293    92A444512Denver, CO 80222                NNYD
4C98894FFFF4444FFCFFFFFFC89A8964CD4FFFFF4444444444444444DDEC444444444444444444
079555012930000921444512455559B0360802220000000000000000558400000000000000000000
 ----------------------------------------------------------------------------
 Watson 1093   92A444513Middlesex,UB11 4AA England      NNNF
4E8AA994FFFF444FFCFFFFFFD88898A8A6ECFF4FCC4C989898444444DDDC444444444444444444
061326501093000921444513494435257B42110411055731540000005556000000000000000000
 ----------------------------------------------------------------------------
```

Figure 124. Sample PBROWSE Subcommand in HEX mode

PGMDOC Subcommand

Purpose The PGMDOC subcommand lists module descriptions from a program table. The output is a one-line description of each module in the member group.

Example PGMDOC   idc*

Syntax

```
PGMDOC memgroup
```

Aliases PG, PGM, PGMD, PGMDO, PGMDOC

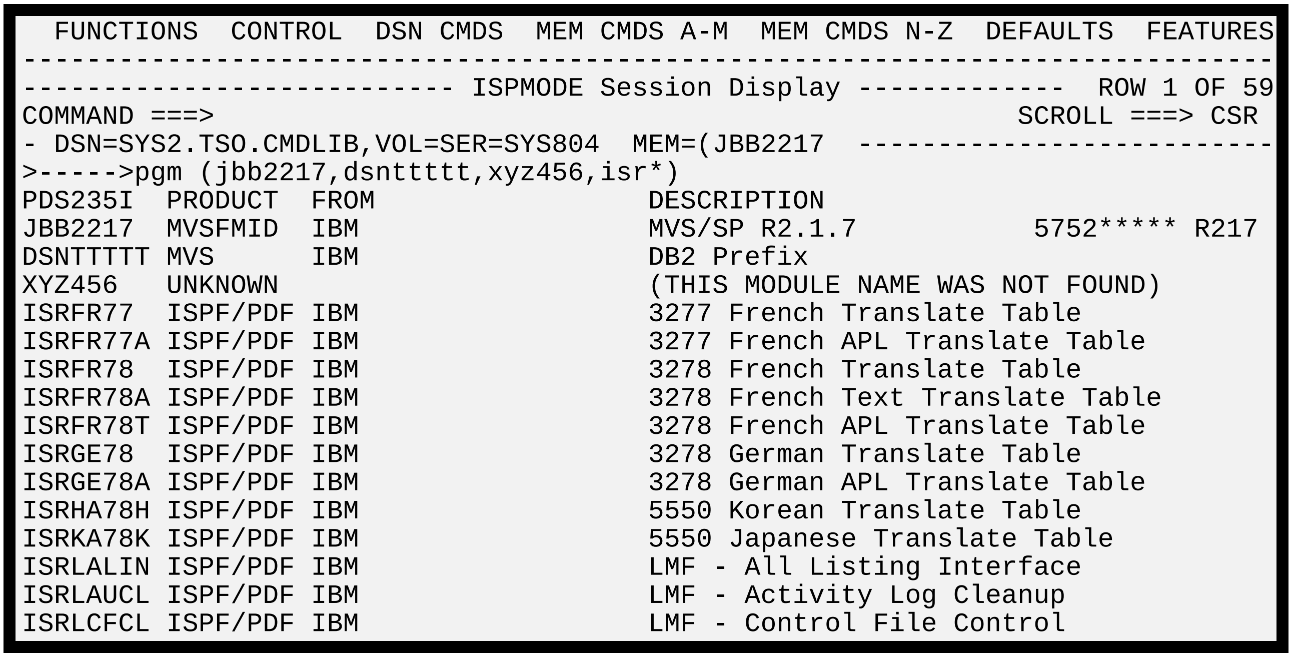
Defaults memgroup

Required none

Operands

memgroup identifies the member name(s) of interest.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
--------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display -------------  ROW 1 OF 59
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SYS2.TSO.CMDLIB,VOL=SER=SYS804  MEM=(JBB2217  --------------------------
>----->pgm (jbb2217,dsnttttt,xyz456,isr*)
PDS235I  PRODUCT  FROM                DESCRIPTION
JBB2217  MVSFMID  IBM                 MVS/SP R2.1.7          5752***** R217
DSNTTTTT MVS      IBM                 DB2 Prefix
XYZ456   UNKNOWN                      (THIS MODULE NAME WAS NOT FOUND)
ISRFR77  ISPF/PDF IBM                 3277 French Translate Table
ISRFR77A ISPF/PDF IBM                 3277 French APL Translate Table
ISRFR78  ISPF/PDF IBM                 3278 French Translate Table
ISRFR78A ISPF/PDF IBM                 3278 French Text Translate Table
ISRFR78T ISPF/PDF IBM                 3278 French APL Translate Table
ISRGE78  ISPF/PDF IBM                 3278 German Translate Table
ISRGE78A ISPF/PDF IBM                 3278 German APL Translate Table
ISRHA78H ISPF/PDF IBM                 5550 Korean Translate Table
ISRKA78K ISPF/PDF IBM                 5550 Japanese Translate Table
ISRLALIN ISPF/PDF IBM                 LMF - All Listing Interface
ISRLAUCL ISPF/PDF IBM                 LMF - Activity Log Cleanup
ISRLCFCL ISPF/PDF IBM                 LMF - Control File Control
```

Figure 130. Sample PGMDOC Subcommand

PRINT Subcommand

Purpose The PRINT subcommand prints a hardcopy list of a member. The TSO PRINTDS or PRINTOFF command is used (as generated during STARWARP installation); any desired PRINTDS/PRINTOFF operands may be added after the member name.

Example PRINT mema:memb form(0012) asa

Syntax

```
PRINT   memgroup [ASA              ]
                 [ASIS/CAPS        ]
                 [CLASS(c)         ]
                 [COPIES(num)      ]
                 [DEST(destname)   ]
                 [FCB(fcbname)     ]
                 [FORMS(formname)  ]
                 [HEADING/NOHEADING]
                 [HOLD/NOHOLD      ]
                 [LIST/NOLIST      ]
                 [NOMSG            ]
                 [PRINT/NOPRINT    ]
                 [PROG(progname)   ]
                 [SNUM             ]
                 [TEXT             ]
                 [UCS(ucsname)     ]
                 [VOLUME(volname)  ]

        Note: the above operands are for the PRINTOFF command.
```

Aliases PR, PRI, PRIN, PRINT

Defaults memgroup, CLASS(A), NOHOLD, LIST, PRINT, ASIS, HEADING

Required none

Operands

memgroup identifies the member(s) to be printed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

If the memgroup is specified as **:** (for all members), the PRINT command will be called only once to print the entire data set.

others optional, may include any desired TSO PRINTOFF or PRINTDS operands.

Remarks Either the TSO PRINTDS or PRINTOFF command is used; this choice was made during STARWARP installation. To determine which program is used, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "PRINT calls". This interface is optional and it should only be used with the TSO PRINTDS command or the public domain PRINTOFF command.

PROFMAN Function

Purpose The PROFMAN (profile manager) command manages saved MEMLIST, LISTC/LISTF and WORKPAD tables. Primary commands such as SORT, F (find), REMOVE and OUTPUT are useful in managing saved tables.

Syntax

```
PROFMAN
```

Aliases PRO, PROF, PROFM, PROFMA, PROFMAN

Defaults none

Required none

Operands none

Remarks PROFMAN builds and displays a table containing all of your save MEMLIST, LISTC/LISTF and WORKPAD tables. From this display, you can inspect, manage and activate any of these tables.

The alternate (reached with a LEFT or RIGHT command) view of this function shows the first line of each saved table. This should give you some context to help you identify a given table.

The following primary commands are supported directly for the PROFMAN function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 91.

ALT[ERNAT] displays an alternate view of the PROFMAN table

DUA[L] displays a double line view of the PROFMAN table.

EDITT[BL] (or **ET[BL]**) enters an edit session on PROFMAN table data.

F finds a string and positions the display start location.
Syntax: `F anystring [ASIS]`
`[ANY/FUNC/NAME/MODIFIED/ID/SIZE/ENTRY]`

LE[FT] rotates through alternate views of the PROFMAN table. PF keys 10 and 22 are normally set to LEFT.

L[OCATE] positions to a data line in sorted tables by searching the current sorted column for the specified data.
Syntax: `LOC {LC/ML/WO[partname]}name/partdate/user/size`

NOR[MAL] displays the default view of the PROFMAN table.

O[PTIONS] provides primary command selection for the PROFMAN function and operand syntax assistance.

OUT[PUT] outputs the PROFMAN table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the PROFMAN table based on a string match.
Syntax: `REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]`
`[ANY/FUNC/NAME/MODIFIED/ID/SIZE/ENTRY]`

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the PROFMAN table. PF keys 11 and 23 are normally set to RIGHT.

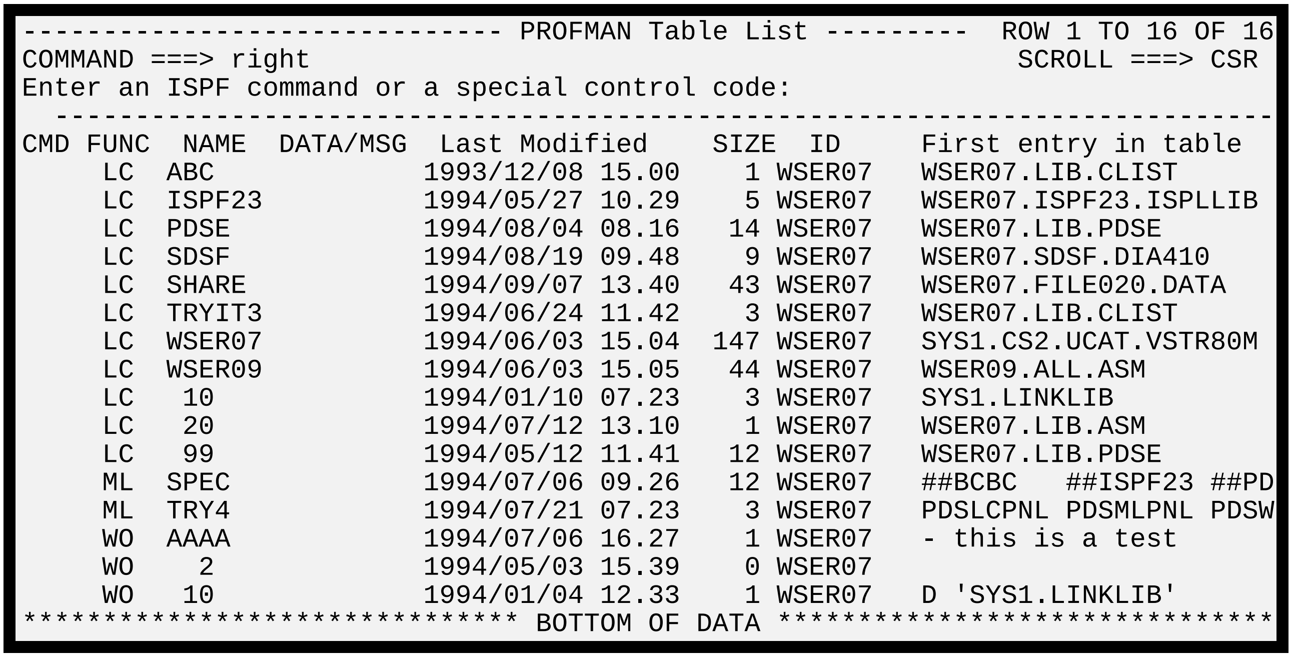
SO[RT] sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order.

PROFMAN Function

	Syntax: `SORT [FUNC/NAME/MODIFIED/ID/SIZE]` `[ASCEND/DESCEND]`
X	clears the PROFMAN relative to the cursor position. Syntax: `X [ABOVE/BELOW/ALL]`
XA[LL]	clears the PROFMAN table; this is equivalent to **X ALL**

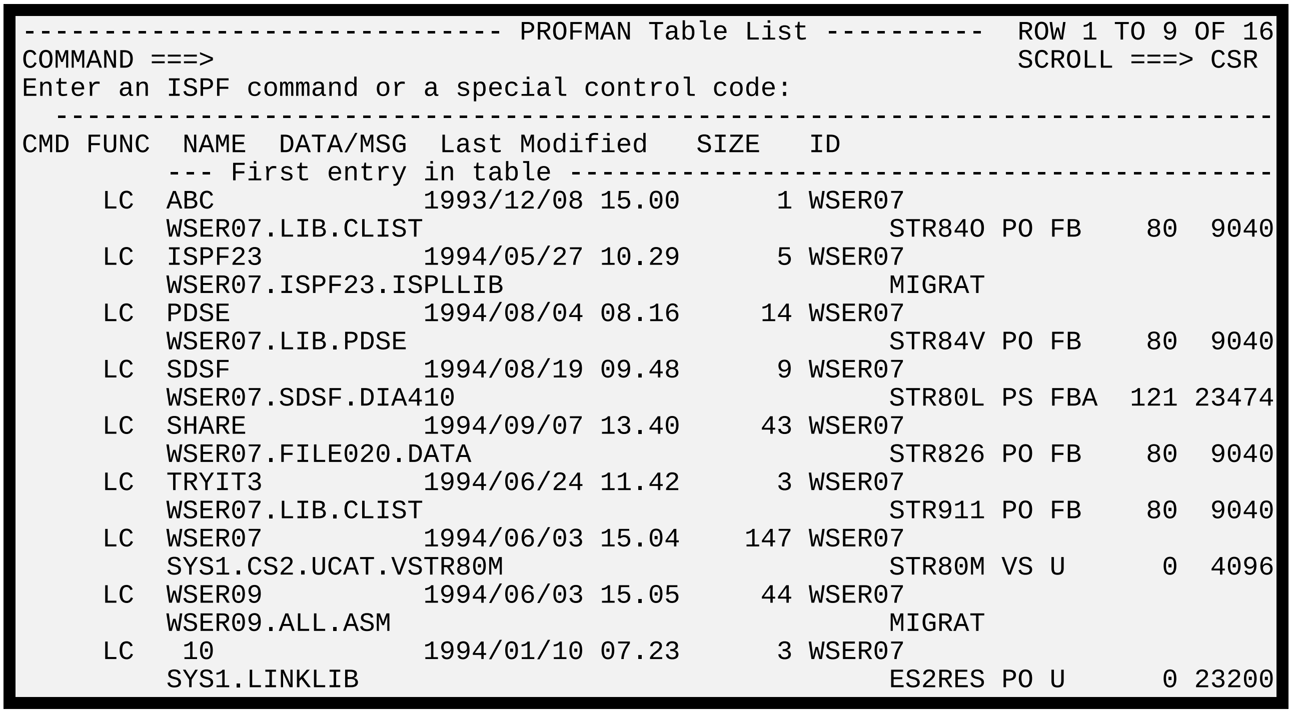
The following line commands are supported in the PROFMAN function.

=	repeat the previous line command.
D	display this saved table.
DEL	delete this saved table.
K	kill and clear all following line commands.
LOG	copy the line into the log.
MERG	merge this table into a MEMLIST, LISTC/LISTF or WORKPAD table
O	provide line command selection and operand syntax assistance.
REN	rename this saved table.
S	select line command (activates the MEMLIST, LISTC/LISTF or WORKPAD table).
X	drop the table line.



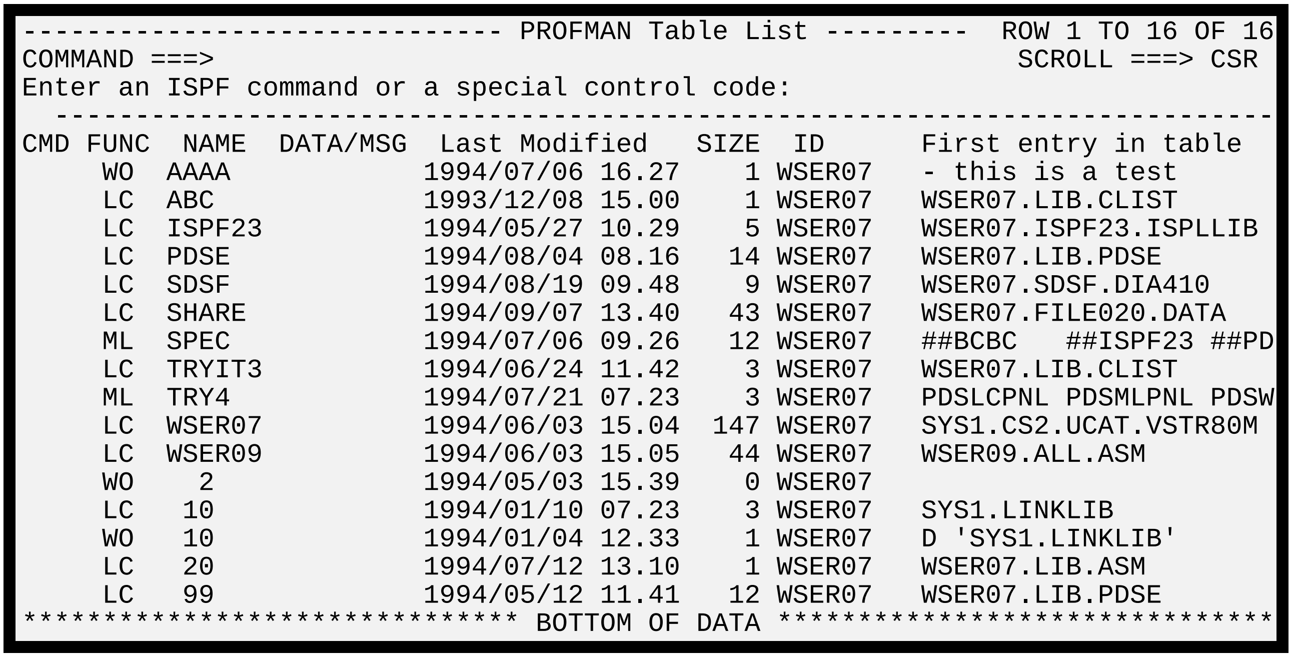
```
------------------------------ PROFMAN Table List ---------  ROW 1 TO 16 OF 16
COMMAND ===> right                                            SCROLL ===> CSR
Enter an ISPF command or a special control code:
  ----------------------------------------------------------------------------
CMD FUNC  NAME  DATA/MSG  Last Modified    SIZE  ID     First entry in table
     LC  ABC             1993/12/08 15.00    1 WSER07   WSER07.LIB.CLIST
     LC  ISPF23          1994/05/27 10.29    5 WSER07   WSER07.ISPF23.ISPLLIB
     LC  PDSE            1994/08/04 08.16   14 WSER07   WSER07.LIB.PDSE
     LC  SDSF            1994/08/19 09.48    9 WSER07   WSER07.SDSF.DIA410
     LC  SHARE           1994/09/07 13.40   43 WSER07   WSER07.FILE020.DATA
     LC  TRYIT3          1994/06/24 11.42    3 WSER07   WSER07.LIB.CLIST
     LC  WSER07          1994/06/03 15.04  147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC  WSER09          1994/06/03 15.05   44 WSER07   WSER09.ALL.ASM
     LC   10             1994/01/10 07.23    3 WSER07   SYS1.LINKLIB
     LC   20             1994/07/12 13.10    1 WSER07   WSER07.LIB.ASM
     LC   99             1994/05/12 11.41   12 WSER07   WSER07.LIB.PDSE
     ML  SPEC            1994/07/06 09.26   12 WSER07   ##BCBC   ##ISPF23 ##PD
     ML  TRY4            1994/07/21 07.23    3 WSER07   PDSLCPNL PDSMLPNL PDSW
     WO  AAAA            1994/07/06 16.27    1 WSER07   - this is a test
     WO    2             1994/05/03 15.39    0 WSER07
     WO   10             1994/01/04 12.33    1 WSER07   D 'SYS1.LINKLIB'
******************************* BOTTOM OF DATA ******************************
```

Figure 131. Sample PROFMAN Function



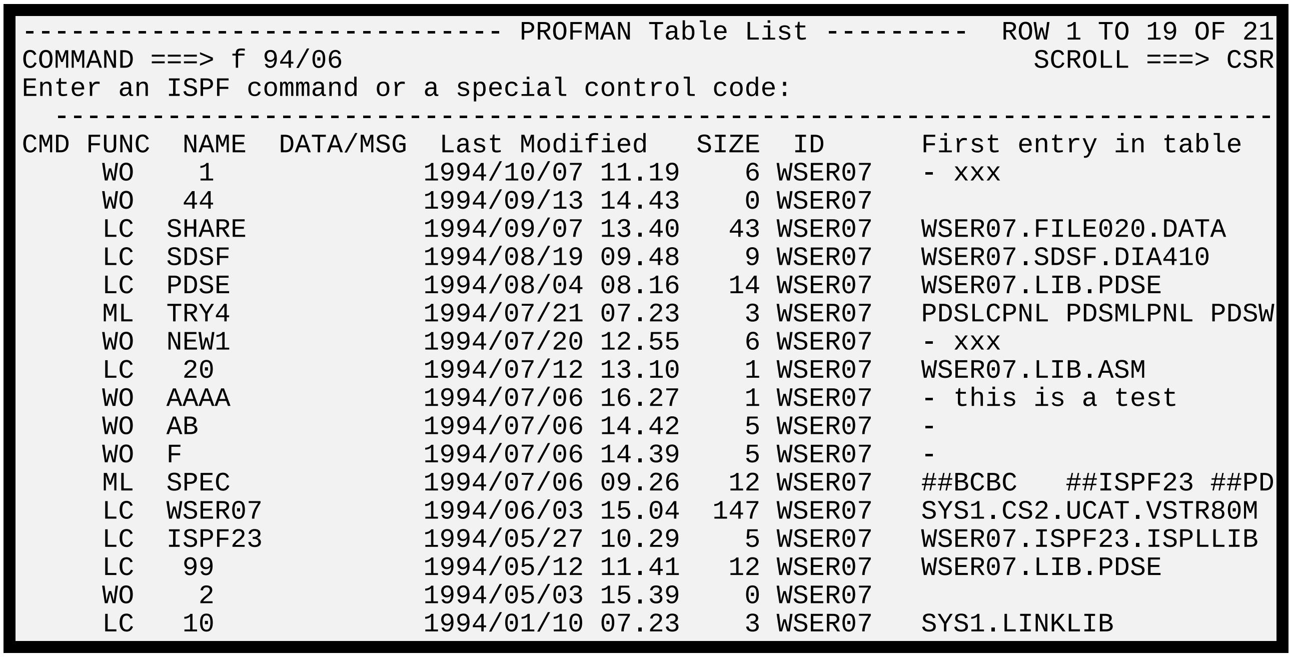
```
------------------------------ PROFMAN Table List ----------  ROW 1 TO 9 OF 16
COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command or a special control code:
  ----------------------------------------------------------------------------
CMD FUNC  NAME  DATA/MSG  Last Modified   SIZE   ID
         --- First entry in table --------------------------------------------
     LC  ABC             1993/12/08 15.00      1 WSER07
         WSER07.LIB.CLIST                             STR84O PO FB    80  9040
     LC  ISPF23          1994/05/27 10.29      5 WSER07
         WSER07.ISPF23.ISPLLIB                        MIGRAT
     LC  PDSE            1994/08/04 08.16     14 WSER07
         WSER07.LIB.PDSE                              STR84V PO FB    80  9040
     LC  SDSF            1994/08/19 09.48      9 WSER07
         WSER07.SDSF.DIA410                           STR80L PS FBA  121 23474
     LC  SHARE           1994/09/07 13.40     43 WSER07
         WSER07.FILE020.DATA                          STR826 PO FB    80  9040
     LC  TRYIT3          1994/06/24 11.42      3 WSER07
         WSER07.LIB.CLIST                             STR911 PO FB    80  9040
     LC  WSER07          1994/06/03 15.04    147 WSER07
         SYS1.CS2.UCAT.VSTR80M                        STR80M VS U      0  4096
     LC  WSER09          1994/06/03 15.05     44 WSER07
         WSER09.ALL.ASM                               MIGRAT
     LC   10             1994/01/10 07.23      3 WSER07
         SYS1.LINKLIB                                 ES2RES PO U      0 23200
```

Figure 132. Sample PROFMAN Right View



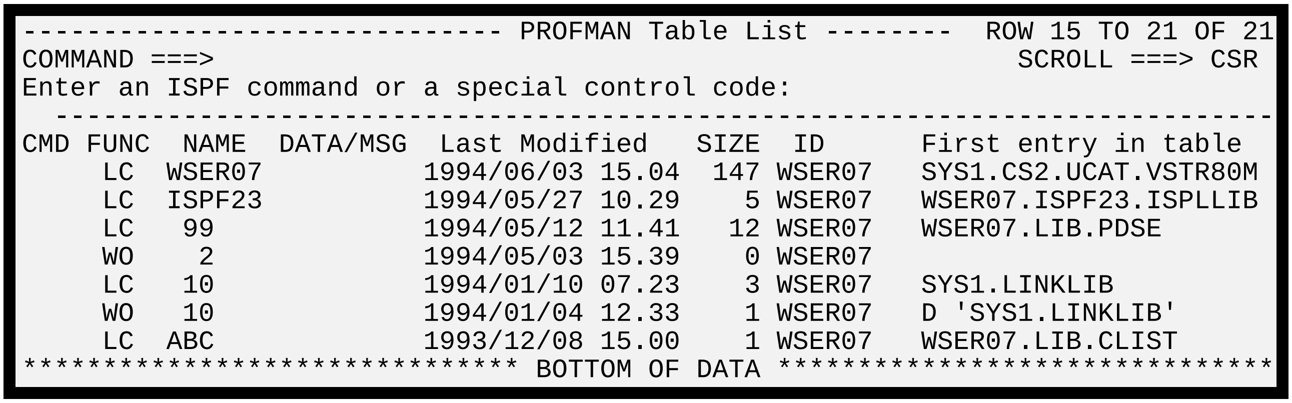
```
------------------------------ PROFMAN Table List ---------  ROW 1 TO 16 OF 16
COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command or a special control code:
  ----------------------------------------------------------------------------
CMD FUNC  NAME  DATA/MSG  Last Modified   SIZE ID      First entry in table
     WO  AAAA            1994/07/06 16.27    1 WSER07   - this is a test
     LC  ABC             1993/12/08 15.00    1 WSER07   WSER07.LIB.CLIST
     LC  ISPF23          1994/05/27 10.29    5 WSER07   WSER07.ISPF23.ISPLLIB
     LC  PDSE            1994/08/04 08.16   14 WSER07   WSER07.LIB.PDSE
     LC  SDSF            1994/08/19 09.48    9 WSER07   WSER07.SDSF.DIA410
     LC  SHARE           1994/09/07 13.40   43 WSER07   WSER07.FILE020.DATA
     ML  SPEC            1994/07/06 09.26   12 WSER07   ##BCBC   ##ISPF23 ##PD
     LC  TRYIT3          1994/06/24 11.42    3 WSER07   WSER07.LIB.CLIST
     ML  TRY4            1994/07/21 07.23    3 WSER07   PDSLCPNL PDSMLPNL PDSW
     LC  WSER07          1994/06/03 15.04  147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC  WSER09          1994/06/03 15.05   44 WSER07   WSER09.ALL.ASM
     WO    2             1994/05/03 15.39    0 WSER07
     LC   10             1994/01/10 07.23    3 WSER07   SYS1.LINKLIB
     WO   10             1994/01/04 12.33    1 WSER07   D 'SYS1.LINKLIB'
     LC   20             1994/07/12 13.10    1 WSER07   WSER07.LIB.ASM
     LC   99             1994/05/12 11.41   12 WSER07   WSER07.LIB.PDSE
******************************* BOTTOM OF DATA *******************************
```

Figure 133. Sample PROFMAN SORT by NAME



```
------------------------------ PROFMAN Table List ---------  ROW 1 TO 19 OF 21
COMMAND ===> f 94/06                                           SCROLL ===> CSR
Enter an ISPF command or a special control code:
  -----------------------------------------------------------------------------
CMD FUNC  NAME  DATA/MSG  Last Modified   SIZE  ID      First entry in table
     WO    1            1994/10/07 11.19    6 WSER07   - xxx
     WO   44            1994/09/13 14.43    0 WSER07
     LC  SHARE          1994/09/07 13.40   43 WSER07   WSER07.FILE020.DATA
     LC  SDSF           1994/08/19 09.48    9 WSER07   WSER07.SDSF.DIA410
     LC  PDSE           1994/08/04 08.16   14 WSER07   WSER07.LIB.PDSE
     ML  TRY4           1994/07/21 07.23    3 WSER07   PDSLCPNL PDSMLPNL PDSW
     WO  NEW1           1994/07/20 12.55    6 WSER07   - xxx
     LC   20            1994/07/12 13.10    1 WSER07   WSER07.LIB.ASM
     WO  AAAA           1994/07/06 16.27    1 WSER07   - this is a test
     WO  AB             1994/07/06 14.42    5 WSER07   -
     WO  F              1994/07/06 14.39    5 WSER07   -
     ML  SPEC           1994/07/06 09.26   12 WSER07   ##BCBC   ##ISPF23 ##PD
     LC  WSER07         1994/06/03 15.04  147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC  ISPF23         1994/05/27 10.29    5 WSER07   WSER07.ISPF23.ISPLLIB
     LC   99            1994/05/12 11.41   12 WSER07   WSER07.LIB.PDSE
     WO    2            1994/05/03 15.39    0 WSER07
     LC   10            1994/01/10 07.23    3 WSER07   SYS1.LINKLIB
```

Figure 134. Sample PROFMAN SORT by MODIFIED



```
------------------------------ PROFMAN Table List --------  ROW 15 TO 21 OF 21
COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command or a special control code:
  -----------------------------------------------------------------------------
CMD FUNC  NAME  DATA/MSG  Last Modified   SIZE  ID      First entry in table
     LC  WSER07         1994/06/03 15.04  147 WSER07   SYS1.CS2.UCAT.VSTR80M
     LC  ISPF23         1994/05/27 10.29    5 WSER07   WSER07.ISPF23.ISPLLIB
     LC   99            1994/05/12 11.41   12 WSER07   WSER07.LIB.PDSE
     WO    2            1994/05/03 15.39    0 WSER07
     LC   10            1994/01/10 07.23    3 WSER07   SYS1.LINKLIB
     WO   10            1994/01/04 12.33    1 WSER07   D 'SYS1.LINKLIB'
     LC  ABC            1993/12/08 15.00    1 WSER07   WSER07.LIB.CLIST
******************************* BOTTOM OF DATA ********************************
```

Figure 135. Sample PROFMAN after F command

QUIT Subcommand

Purpose The QUIT subcommand terminates STARWARP.

Example QUIT

Syntax

```
QUIT
```

Aliases Q, QU, QUI, QUIT

Operands (no operands are supported for the QUIT subcommand).

Remarks The QUIT subcommand is equivalent to multiple END subcommands. It is normally used from ISPMODE or MEMLIST instead of repeated ENDs when you want to terminate STARWARP and not just the current function.

READOBJ Subcommand

Purpose The READOBJ subcommand reads and disassembles object code.

Example READOBJ PTF1107

Syntax

```
READOBJ memgroup [FLOAT/NOFLOAT         ]
                 [MVS370/MVSXA/ESA370   ]
                 [PRIV/NOPRIV           ]
                 [REASM                 ]
```

Aliases READ, READO, READOB, READOBJ

Defaults memgroup, NOFLOAT, NOPRIV, ESA370

Required none

Operands

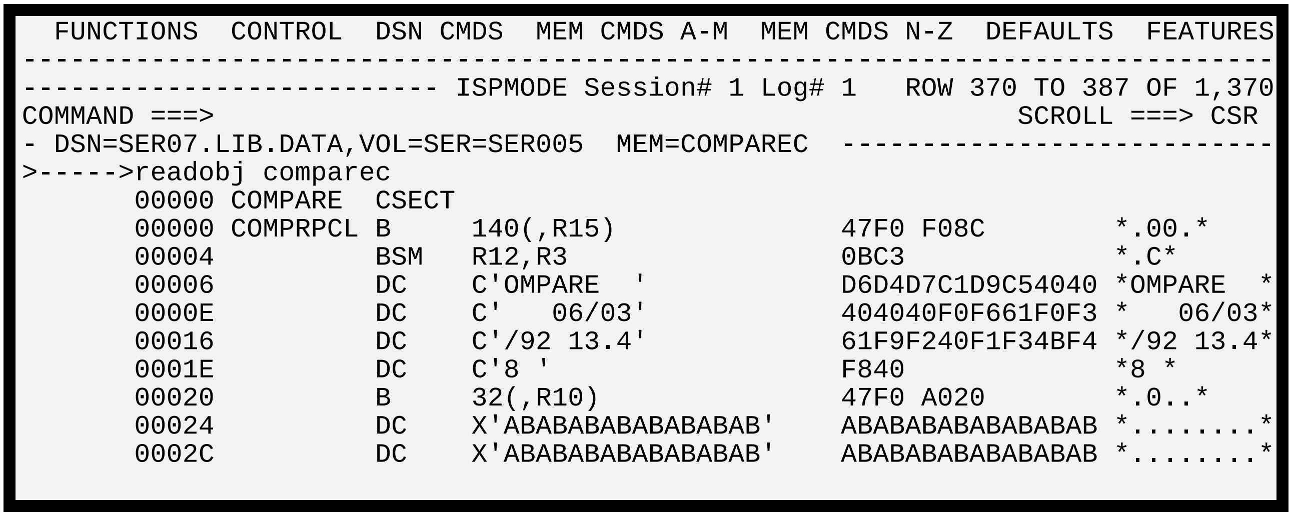
memgroup identifies the member(s) to be disassembled.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

FLOAT include floating-point instructions in the instruction set.
NOFLOAT do not decode floating-point instructions.
MVS370 use only instructions defined on MVS/370 systems.
MVSXA use only instructions defined on MVS/XA systems.
ESA370 use only instructions defined on MVS/ESA systems; this instruction set also includes all MVS/XA instructions.
PRIV include privileged instructions in the instruction set.
NOPRIV do not decode privileged instructions.
REASM output the disassembled instructions in a format which can be reassembled to recreate the decoded module. When this format is requested, the CSECT offset field (the first output field) is placed at the end of each decoded statement.

Remarks READOBJ uses an algorithm that should allow several object decks to be present in a single member. READOBJ formats its output as follows:

CSECT offset a hexadecimal offset in the current CSECT. Note: with REASM format, this would be the last output field.
name a external name or generated internal name (format: "A" followed by a six character offset in the module).
operation a one to five byte reconstructed operation code.
operands the reconstructed operand values.
comments for SVC calls, the original MACRO (for example, GETMAIN or TPUT).
hex representation one to three columns of the original hexadecimal data.
character representation one to 8 bytes of the same data in character mode surrounded by asterisks.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1   ROW 370 TO 387 OF 1,370
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.DATA,VOL=SER=SER005  MEM=COMPAREC  ---------------------------
>----->readobj comparec
       00000 COMPARE  CSECT
       00000 COMPRPCL B     140(,R15)              47F0 F08C        *.00.*
       00004          BSM   R12,R3                 0BC3             *.C*
       00006          DC    C'OMPARE  '            D6D4D7C1D9C54040 *OMPARE  *
       0000E          DC    C'   06/03'            404040F0F661F0F3 *   06/03*
       00016          DC    C'/92 13.4'            61F9F240F1F34BF4 */92 13.4*
       0001E          DC    C'8 '                  F840             *8 *
       00020          B     32(,R10)               47F0 A020        *.0..*
       00024          DC    X'ABABABABABABABAB'    ABABABABABABABAB *........*
       0002C          DC    X'ABABABABABABABAB'    ABABABABABABABAB *........*
```

Figure 136. Sample READOBJ Subcommand

RECALL Subcommand

Purpose The RECALL (or RC) subcommand displays the previous subcommand or displays and reenters the previous subcommand for execution.

Note: the following subcommands are ignored for the purposes of the RECALL subcommand: CHANGE, HELP, K (KLEAR) and TSO.

Example RECALL enter

Syntax

```
RECALL   [ENTER/NOENTER/CHANGE]
```

Aliases R, RC, RE, REC, RECA, RECAL, RECALL

Defaults NOENTER

Required none

Operands

ENTER display and reenter the previous subcommand.
NOENTER display the previous subcommand for overtyping.
CHANGE generate a prototype CHANGE subcommand for the currently allocated data set.

Remarks In ISPMODE additional options are available to allow you to retrieve subcommands from the log; also, LASTCMDS allows you to display or select any of the last 32 primary subcommands entered.

REPLACE Subcommand

Purpose The REPLACE subcommand changes portions of a member which contain a search string. Only those portions of the member which contain the search string are displayed after a character string substitution. For load modules or VSAM data sets, the search and replace strings must have the same length.

Example REPLACE mema:memb /before/after/ write
   (or equivalently)
REPLACE mema:memb 'before' 'after' write
   (or equivalently)
REPLACE mema:memb xc2c5c6d6d9c5xc1c6e3c5d9x write

Syntax

```
REPLACE   memgroup
          +stra+strb+
          [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP  ]
          [CAPS/ASIS/IGNORE/PICTURE                ]
          [STATS/NOSTATS                           ]
          [WRITE/NOWRITE/UPDATE/NOUPDATE           ]
          [WORD/PREFIX/SUFFIX                      ]
          [OFFSET(Hx) MODULE({* / Fullm / Partm*)} ]
          [AFTER(num) DO(num) MAXMEMBERS(num) COLS(from:to) ]
          [AND('string2') ACOLS(from:to) ACAPS/AASIS/AIGNORE/APICTURE]
          [OR('string3') OCOLS(from:to) OCAPS/OASIS/OIGNORE/OPICTURE ]
          [FORMAT(from:to,from:to, ...) ]
          [SKIPREC(n) MAXIN(n) MAXOUT(n)           ]
          [SKIPCOL(n) MAXLEN(n) MAXFIND(n)         ]
          [MEMBERS / MEMLIST / ML / NEWML / SUBLIST]
        * [FROMKEY(key)/FROMADDRESS(add)/FROM (num)]
        * [TOKEY(key)/TOADDRESS(add)/TONUMBER      ]
*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases REP, REPL, ,REPLA, REPLAC, REPLACE

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults memgroup, previous stringa, IGNORE, NOWRITE, STATS, NUM or previous REPLACE/FIND/LIST format

Required none

Operands

memgroup identifies the member(s) to be scanned and updated.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

stringa defines the string to be searched for. It is coded as a delimited string like **+searched for+**. If the string is not entered or is null, the last string entered will be used. The string may contain one to 32 characters. If the ASIS keyword is entered, neither string will be translated to upper case letters.

For LBLOCK, LDUMP, BLOCK and DUMP formats, the search and replace strings must have the same lengths.

As an alternative, both strings may be entered as hexadecimal strings delimited with x's like **x0123456789xabcdefx**. Note that each string may contain one to 64 characters and that x333x1x and x0333x01x are equivalent.

stringb defines the replacement string. It is coded as a delimited string like **+replaced with+**. If the string is not entered or is null, the string is assumed to be null. The string may contain zero to 32 characters. If the ASIS keyword is entered, neither string will be translated to upper case letters.

For LBLOCK, LDUMP, BLOCK and DUMP formats, the search and replace strings must have the same lengths.

As an alternative, both strings may be entered as hexadecimal strings delimited with x's like **x0123456789xabcdefx**. Note that each string may contain one to 64 characters and that x333x1x and x0333x01x are equivalent.

NUM examine the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise) of each logical record; if the line number field is not numeric, switch formatting to NONUM mode for the remainder of the member.

For ISPF-saved members, the high-order six digits of the line number field is formatted; otherwise, the low-order six digits of the line number field is formatted by suppressing leading zeroes. The line number segment is followed by a blank and up to 249 characters of data from the logical record.

SNUM discard the line number field (the last 8 bytes of a fixed-format record or the first 8 bytes otherwise); search and update up to 256 bytes from a logical record.

NONUM search or update up to 256 bytes from a logical record (without regard to line numbers).

LBLOCK format logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record) and segments of up to 64 characters surrounded by asterisks. Only formatted segments which contain the search string are updated and displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for load modules, only CSECT data will be displayed or updated; if a name is entered in a MODULE operand, only those CSECT or ENTRY names which satisfy the condition will be displayed or updated. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset) and segments of up to 64 characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a + will be placed after the key); otherwise it is a relative byte address.

LDUMP format logical records with two hexadecimal offsets (displacement in the member and displacement in the logical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks. Only formatted segments which contain the search string are updated and displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for load modules, only CSECT data will be displayed or updated; if a name is entered in a MODULE operand, only those CSECT or ENTRY names which satisfy the condition will be displayed or updated. CSECT records are formatted with two hexadecimal offsets (a module offset and a CSECT offset), segments of up to 32 characters of hexadecimal data and up to 16 bytes of characters surrounded by asterisks.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a **+** will be placed after the key); otherwise it is a relative byte address.

BLOCK format physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record) and segments of up to 64 characters surrounded by asterisks.

Only formatted segments which contain the search string are updated and displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a **+** will be placed after the key); otherwise it is a relative byte address.

If control interval processing is being performed, the first field is always the control interval relative byte address.

DUMP format physical records with two hexadecimal offsets (displacement in the member and displacement in the physical record), segments of up to 32 characters of hexadecimal data and up to 16 bytes of equivalent characters surrounded by asterisks.

Only formatted segments which contain the search string are updated and displayed; however, a following segment will also be displayed if the string spans a segment boundary.

Note: for VSAM records, the first field in the output is the relative record number for a fixed or variable RRDS; up to eight bytes of the key for a KSDS (if the key is longer than eight characters, a **+** will be placed after the key); otherwise it is a relative byte address.

If control interval processing is being performed, the first field is always the control interval relative byte address.

CAPS translate character string arguments to upper case letters.

ASIS do not translate character string arguments to upper case letters.

IGNORE search for upper and lower-case string arguments.

PICTURE if specified, search for data using matching characters and the following picture characters:

=	for any character
%	for any alphanumeric character
@	for any alphabetic character
#	or any numeric character
$	for any special character
~	for any non-blank character (note: the “not” character can also be used)
.	for any any invalid character
-	for any any non-numeric character
<	for any lower case alphabetic character
>	for any upper case alphabetic character

Replacement data may use characters and the following special picture characters:

=	for any character
<	for lower case translation
>	for upper case translation

WORD search for strings preceded and followed by a non-alphameric character. This can be used to limit the string hits.

PREFIX search for strings preceded by a non-alphameric character. This can be used to limit the string hits.

SUFFIX search for strings followed by a non-alphameric character. This can be used to limit the string hits.

WRITE update the member; this is an update-in-place operation.

NOWRITE do not update the member; this is a trial update operation.

UPDATE update the member; this is an update-in-place operation.

NOUPDATE do not update the member; this is a trial update operation.

STATS update ISPF statistics or a ZAP IDR record if any member data is changed.

NOSTATS do not update ISPF statistics or a ZAP IDR record even if some data is changed.

OFFSET(Hx) specifies a 1 to 6 digit hexadecimal module offset at which the search is to begin. Note: this operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT or ENTRY names to be searched or updated. Note: this operand is only defined for load modules and only with the LDUMP or LBLOCK output formats.

If both MODULE and OFFSET keywords are entered, the offset applies to each module selected. The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.

MODULE(Fullm) search or update only a CSECT or ENTRY named FULLM.

MODULE(Partm*) search or update only a CSECT or ENTRY named PARTM...

AFTER(n) is coded as 0 through 9999999 to specify the number of matches to skip before performing replacements.

MAXMEMBERS(n) n is coded as 0 through 9999999 to specify the maximum number of members to be selected by REPLACE.

DO(n) is coded as 0 through 9999999 to specify the maximum number of lines to output for a member.

COLS(from:to) column range to search for the primary string (COLS is only supported for formats NUM, SNUM and NONUM). For example, **COLS(1:10)** specifies that the string may begin in columns 1 through 10.

AND('str2') another string to search in addition to the primary string. AND strings are only supported for formats NUM, SNUM or NONUM and a REPLACE will be reported for a primary string only if the AND string is found first.

ACOLS(from:to) column range to search for the AND string. For example, **ACOLS(1:10)** specifies that the string may begin in columns 1 through 10.

ACAPS if an AND character string argument is used, it is to be translated to upper case letters.

AASIS if an AND character string argument is used, it is not to be translated to upper case letters.

AIGNORE if an AND character string argument is used, search for upper and lower-case data matching the string.

APICTURE if an AND character string argument is used, search for data with matching characters and the following picture characters:

=	for any character
%	for any alphanumeric character
@	for any alphabetic character
#	or any numeric character
$	for any special character
~	for any non-blank character (note: the “not” character can also be used)
.	for any any invalid character
-	for any any non-numeric character
<	for any lower case alphabetic character
>	for any upper case alphabetic character

OR('str3') another string to search as an alternate to the primary string. OR strings are only supported for formats NUM, SNUM or NONUM and a REPLACE will be reported for either a primary string or an OR string.

OCOLS(from:to) column range to search for the OR string. For example, **OCOLS(1:10)** specifies that the string may begin in columns 1 through 10.

OCAPS if an OR character string argument is used, it is to be translated to upper case letters.

OASIS if an OR character string argument is used, it is not to be translated to upper case letters.

OIGNORE if an OR character string argument is used, search for upper and lower-case data matching the string.

OPICTURE if an OR character string argument is used, search for data with matching characters and the following picture characters:

=	for any character
%	for any alphanumeric character
@	for any alphabetic character
#	or any numeric character
$	for any special character
~	for any non-blank character (note: the “not” character can also be used)
.	for any any invalid character
-	for any any non-numeric character
<	for any lower case alphabetic character
>	for any upper case alphabetic character

FORMAT(from:to,from:to, ...) specifies the record columns to display in the output when reporting a REPLACE string. FORMAT is only supported with NUM, SNUM or NONUM data. For example, **FORMAT(21:30,0,1:10)** specifies that the output should be formatted with data from columns 21 through 30, a blank and data from columns 1 through 10.

SKIPREC(n) ignore **n** (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) at the beginning of a member.

MAXIN(n) input up to **n** (coded as 0 through 9999999) logical records (physical records for BLOCK or DUMP formats) for a member after satisfying any SKIPREC operand.

MAXOUT(n) display up to **n** (coded as 0 through 9999999) output lines for a member.

MAXFIND(n) locate and update up to **n** (coded as 0 through 9999999) strings in a member.

SKIPCOL(n) ignore **n** (coded as 0 through 99999) columns at the beginning of each logical record (physical record for BLOCK or DUMP formats).

Note: for NUM or SNUM output format with record format V, SKIPCOL(0) refers to the first data position after the line number field.

MAXLEN(n) search or update up to **n** (coded as 0 through 99999) characters in a logical record (physical record for BLOCK or DUMP formats).

MEMBERS displays the names of members which contain the find string without changing the current member group.

MEMLIST Same as **ML**. Specifies that any member containing the find string will be selected for MEMLIST display after all REPLACE activity is complete. If no members are selected, a null sublist is the result.

ML Same as **MEMLIST**. Specifies that any member containing the find string will be selected for MEMLIST display after all REPLACE activity is complete. If no members are selected, a null sublist is the result.

NEWML Same as **MEMLIST** and **ML** except that the current MEMLIST is reset.

SUBLIST Specifies that any member containing the find string will be selected for inclusion in a new sublist after all REPLACE activity is complete. If no members are selected, a null sublist is the result.

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access begins at the first record whose key matches (or is greater than) the portion of the key specified.

This parameter may be used with TOKEY and it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can be specified for any VSAM data set component if control interval processing is being used.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.

This parameter may be used with TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified.

This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record (or a control interval for control interval processing). If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can be specified for any VSAM data set component if control interval processing is being used.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.

This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

Remarks If the WRITE or UPDATE keyword is specified, the data is updated in-place; otherwise, only potential changes are listed.

Unprintable characters for 3270-type devices are translated to periods before they are displayed.

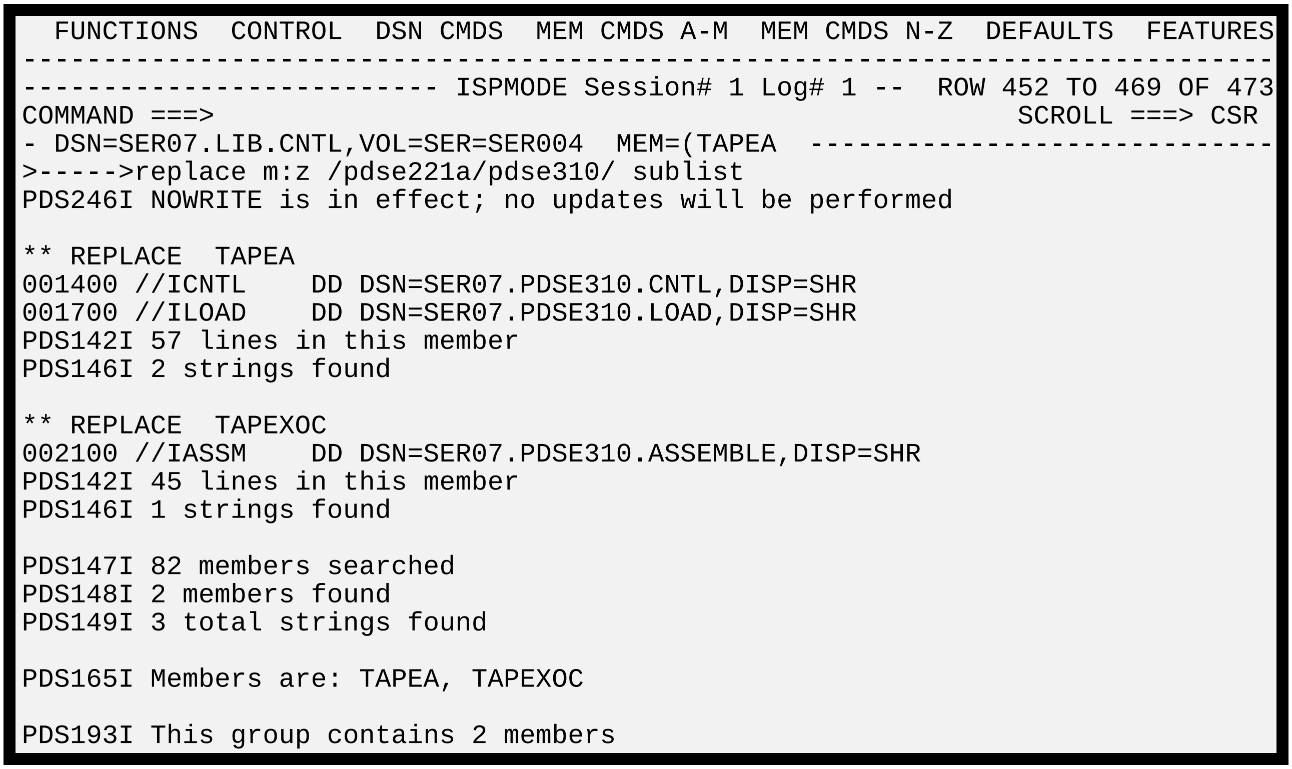
REPLACE formats are NUM, SNUM, NONUM, LBLOCK, LDUMP, BLOCK and DUMP; for load modules, the NUM, SNUM and NONUM formats are equivalent to the LDUMP format and for VSAM data sets, the NUM, SNUM and NONUM formats are equivalent to the LBLOCK format.

The default format is initially NUM; however, each time a format operand is entered on a REPLACE, FIND or LIST subcommand, the value entered is used as the output format for subsequent REPLACE, FIND and LIST subcommands. Also, note the following:

1. Formats NUM, SNUM and NONUM limit the search and update length for logical records to 256 characters.
2. Column 72 is not searched or modified for formats NUM or SNUM if the record format is fixed with 80 character records as this is normally the "continuation" column.
3. Formats BLOCK and DUMP apply to physical records for non-VSAM data sets (or when control interval processing is being performed); the other formats apply to logical records. For load modules, LBLOCK and LDUMP formats update and display only CSECT data.
4. Formats LBLOCK, LDUMP, BLOCK and DUMP display only the updated segments of a record which contained the search string; however, a following segment will also be displayed if the string spans a segment boundary.
5. Formats LBLOCK, LDUMP, BLOCK and DUMP require equal length search and replacement strings.

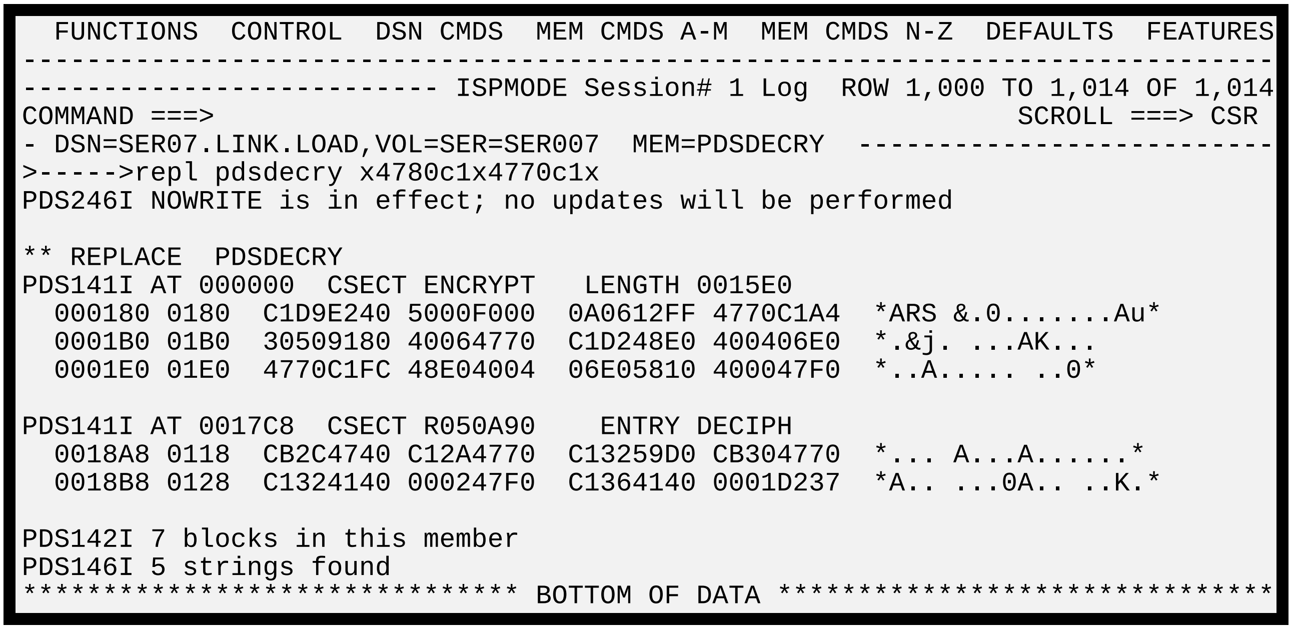
6. Formats NUM, SNUM and NONUM allow different length strings with string expansion (shifting the data over to multiple blanks) and string contraction (shifting the data up to the next blank into the replacement data). The string expansion and contraction algorithms will normally modify programming language (Assembler, CLIST, COBOL, FORTRAN, ISPF, PL/I, ...) statements correctly.
7. For VSAM DATA or INDEX components, the LIST, FIND and REPLACE subcommands support control interval access using the DUMP or BLOCK display formats. Instead of accessing individual VSAM records, each GET or PUT obtains a VSAM control interval.

   Control interval access could be useful if a VSAM data set has logical errors. REPLACE could be used to repair the error; however, since only the component is opened for update, the next access of the data set through the related cluster will get warning errors due to the differing time stamps.



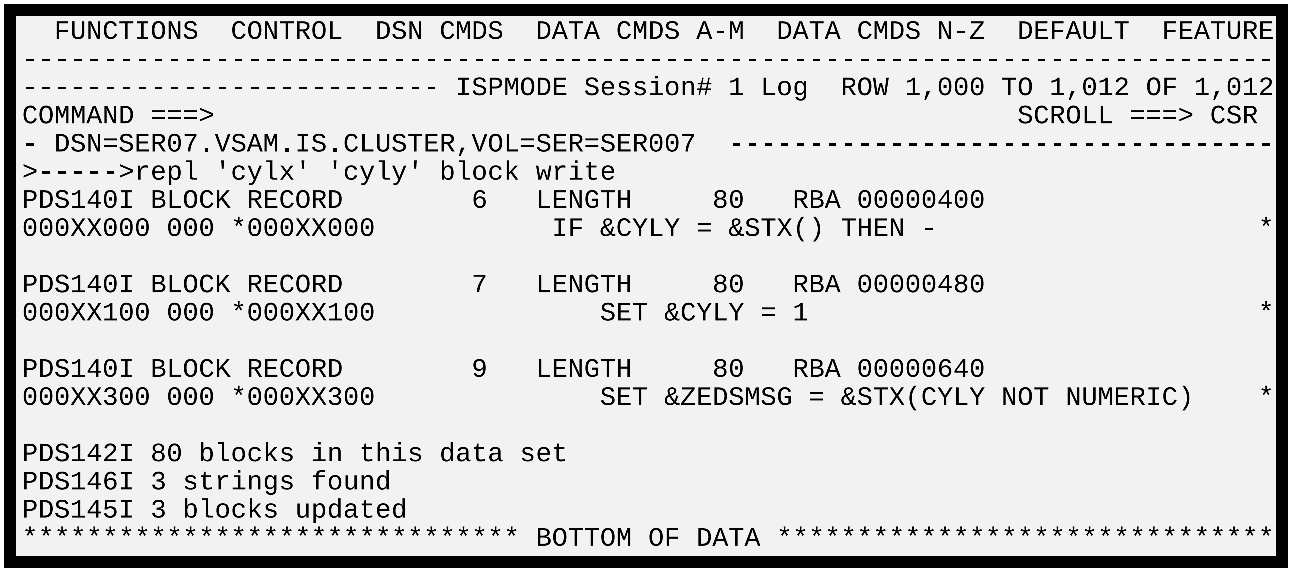
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
-------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1 --  ROW 452 TO 469 OF 473
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER004  MEM=(TAPEA  -----------------------------
>----->replace m:z /pdse221a/pdse310/ sublist
PDS246I NOWRITE is in effect; no updates will be performed

** REPLACE  TAPEA
001400 //ICNTL    DD DSN=SER07.PDSE310.CNTL,DISP=SHR
001700 //ILOAD    DD DSN=SER07.PDSE310.LOAD,DISP=SHR
PDS142I 57 lines in this member
PDS146I 2 strings found

** REPLACE  TAPEXOC
002100 //IASSM    DD DSN=SER07.PDSE310.ASSEMBLE,DISP=SHR
PDS142I 45 lines in this member
PDS146I 1 strings found

PDS147I 82 members searched
PDS148I 2 members found
PDS149I 3 total strings found

PDS165I Members are: TAPEA, TAPEXOC

PDS193I This group contains 2 members
```

Figure 138. Sample REPLACE (source)



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,014 OF 1,014
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LINK.LOAD,VOL=SER=SER007  MEM=PDSDECRY  --------------------------
>----->repl pdsdecry x4780c1x4770c1x
PDS246I NOWRITE is in effect; no updates will be performed

** REPLACE  PDSDECRY
PDS141I AT 000000  CSECT ENCRYPT   LENGTH 0015E0
  000180 0180  C1D9E240 5000F000  0A0612FF 4770C1A4  *ARS &.0.......Au*
  0001B0 01B0  30509180 40064770  C1D248E0 400406E0  *.&j. ...AK...
  0001E0 01E0  4770C1FC 48E04004  06E05810 400047F0  *..A..... ..0*

PDS141I AT 0017C8  CSECT R050A90    ENTRY DECIPH
  0018A8 0118  CB2C4740 C12A4770  C13259D0 CB304770  *... A...A......*
  0018B8 0128  C1324140 000247F0  C1364140 0001D237  *A.. ...0A.. ..K.*

PDS142I 7 blocks in this member
PDS146I 5 strings found
****************************** BOTTOM OF DATA ******************************
```

Figure 139. Sample REPLACE (load)



```
  FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,012 OF 1,012
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.VSAM.IS.CLUSTER,VOL=SER=SER007  ----------------------------------
>----->repl 'cylx' 'cyly' block write
PDS140I BLOCK RECORD        6   LENGTH     80   RBA 00000400
000XX000 000 *000XX000           IF &CYLY = &STX() THEN -                    *

PDS140I BLOCK RECORD        7   LENGTH     80   RBA 00000480
000XX100 000 *000XX100              SET &CYLY = 1                            *

PDS140I BLOCK RECORD        9   LENGTH     80   RBA 00000640
000XX300 000 *000XX300              SET &ZEDSMSG = &STX(CYLY NOT NUMERIC)    *

PDS142I 80 blocks in this data set
PDS146I 3 strings found
PDS145I 3 blocks updated
****************************** BOTTOM OF DATA ******************************
```

Figure 140. Sample REPLACE (VSAM KSDS)

REVIEW Subcommand

Purpose The REVIEW subcommand browses data. The TSO REVIEW command is used; any desired REVIEW operands may be added after the member name. Note: REVIEW operates independently of ISPF.

Example REVIEW   mema:memb

Syntax

```
REVIEW   memgroup [operands]
```

Aliases REV, REVI, REVIE, REVIEW

Defaults none

Required none

Operands

memgroup identifies the member(s) to be browsed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

If the memgroup is specified as **:** (for all members), the REVIEW command will be called to provide a member selection list.

operands optional, may include any desired TSO REVIEW operands.

Remarks This interface is optional, it should only be used if your installation has installed the public domain REVIEW command.

Note that recent versions of the REVIEW command operate in seven-color mode and REVIEW can display VSAM data sets. In a MEMLIST, REV may be used as a line command to display deleted members before they are restored.

SUBLIST Subcommand

Purpose

The SUBLIST subcommand forms lists (or sublists) of members. When SUBLIST gains control, it forms a new default member group using the individual member names passed to it; each subcommand which invokes SUBLIST defines a new member sublist.

For example, if a data set contains members MEMABC, MEMXYZ, MEM33 and MEZ4, entering the command **SUBLIST MEM*** would change the PDS300A message MEM= keyword to **MEM=(MEMABC**. The default member group would be the list of individual member names MEMABC, MEMXYZ, MEM33 and MEZ4 instead of the rule, MEM*, that is usual with STARWARP subcommands.

Example

```
IF amx/   noalias  then(sublist)
PRINT  *  form(dx00)  nohead
COPY  *  other.library  alias
DELETE  *  alias
```

Syntax

```
SUBLIST   memgroup [ALIAS/NOALIAS       ]
                   [REVERSE             ]
                   [EXCLUDE(mem*name)   ]
                   [ONLYALIAS/ONLYMAIN  ]
                   [NOBLDL              ]
```

Aliases

SUBL, SUBLI, SUBLIS, SUBLIST

Defaults

memgroup, NOALIAS

Required

none

Operands

memgroup identifies the group of members whose names are to be defined as a list of members.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

ALIAS also add any associated members (aliases, apparent aliases and associated main members) to the sublist of members.

NOALIAS add only specifically identified member names to the sublist of members.

REVERSE invert the member list to include members from the data set but not in the member list.

EXCLUDE(mem*name) drop members matching the combination name from the sublist.

ONLYALIAS forces ALIAS on and adds associated members (aliases, apparent aliases and associated main members) to the sublist of members; then, the main members are dropped from the sublist leaving "only aliases".

ONLYMAIN forces ALIAS on and adds associated members (aliases, apparent aliases and associated main members) to the sublist of members; then, the alias members are dropped from the sublist leaving "only main members".

NOBLDL specifies that the existence of each member in the current group need not be verified with a BLDL macro.

This parameter is intended for high performance applications in conjunction with the MEMBERS subcommand.

Note that this keyword permits processing a list of members without discarding members which are not present

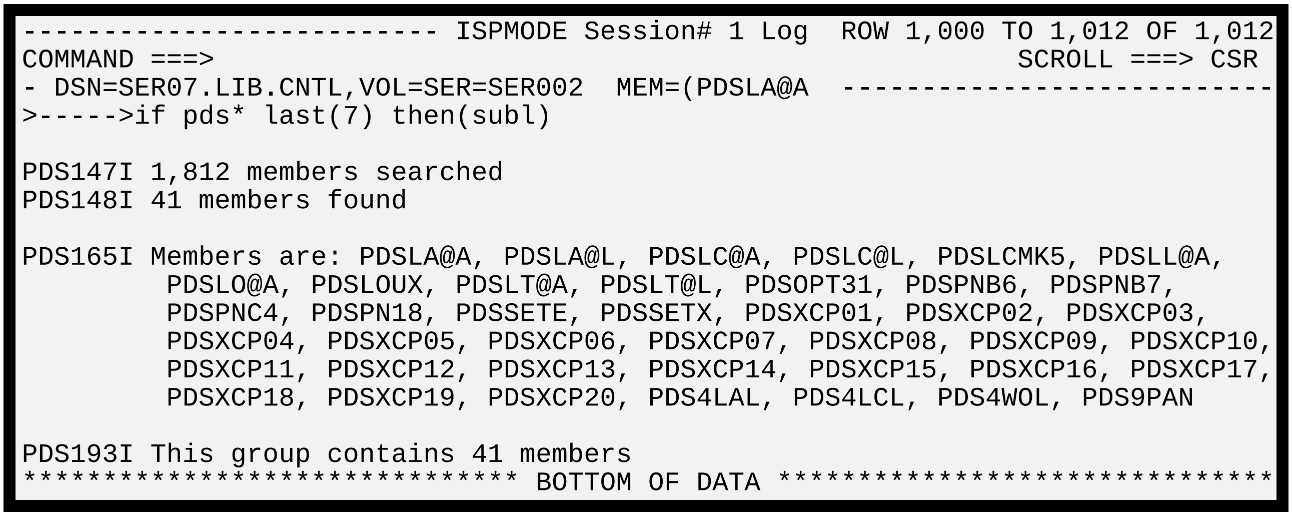
Remarks The SUBLIST subcommand disregards duplicate member names.

If multiple operations are to be performed to the same list of members, they can be organized as a single member group using the SUBLIST subcommand.

SUBLIST is useful for the THEN or ELSE keywords of the IF and FIND subcommands. Operands are not supported after the subcommand name (in the THEN or ELSE operand). However, if SUBLIST is used as the subcommand, a following subcommand (which uses * for the group name) may have any desired operands.

SUBLIST is also useful for the THEN or ELSE keywords of the IF and FIND subcommands for situations where a refined sublist is being built. For example, to print all members named A11...XM or MEZ... with ISPF statistics which were saved in the last year and which do not contain the character string "NOTME", the following STARWARP subcommands would suffice:

```
if  (a11*xm,mez*)  year  then(sublist)
find  *  'notme'  else(sublist)
print  *
```



```
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,012 OF 1,012
COMMAND ===>                                                   SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER002  MEM=(PDSLA@A  --------------------------
>----->if pds* last(7) then(subl)

PDS147I 1,812 members searched
PDS148I 41 members found

PDS165I Members are: PDSLA@A, PDSLA@L, PDSLC@A, PDSLC@L, PDSLCMK5, PDSLL@A,
         PDSLO@A, PDSLOUX, PDSLT@A, PDSLT@L, PDSOPT31, PDSPNB6, PDSPNB7,
         PDSPNC4, PDSPN18, PDSSETE, PDSSETX, PDSXCP01, PDSXCP02, PDSXCP03,
         PDSXCP04, PDSXCP05, PDSXCP06, PDSXCP07, PDSXCP08, PDSXCP09, PDSXCP10,
         PDSXCP11, PDSXCP12, PDSXCP13, PDSXCP14, PDSXCP15, PDSXCP16, PDSXCP17,
         PDSXCP18, PDSXCP19, PDSXCP20, PDS4LAL, PDS4LCL, PDS4WOL, PDS9PAN

PDS193I This group contains 41 members
******************************* BOTTOM OF DATA *******************************
```

Figure 149. Sample SUBLIST Subcommand

SUBMIT Subcommand

Purpose The SUBMIT subcommand submits a JCL member. The TSO SUBMIT command is used; any desired SUBMIT operands may be specified after the member name.

Example SUBMIT   mema:memb

Syntax

```
SUBMIT   memgroup [operands]
```

Aliases SU, SUB, SUBM, SUBMI, SUBMIT

Defaults memgroup

Required none

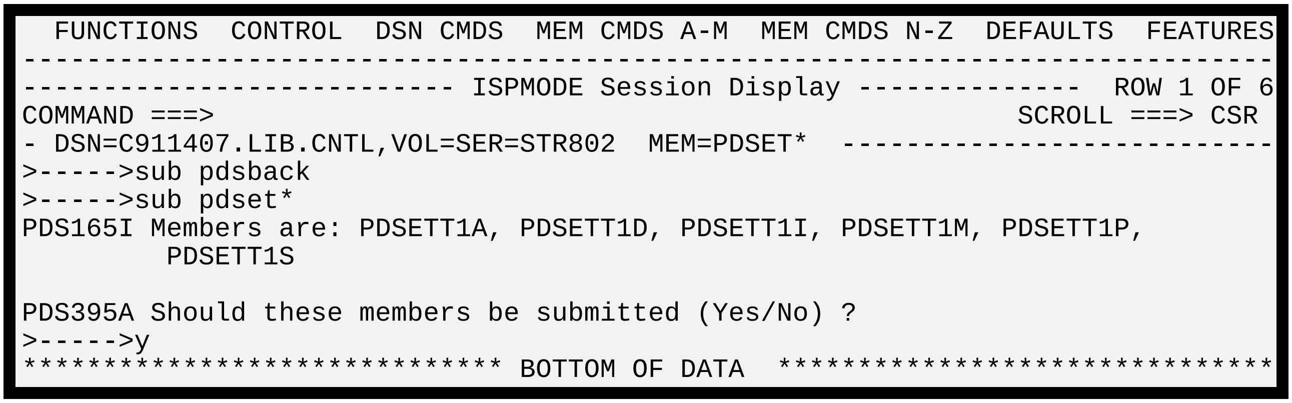
Operands

memgroup identifies the member(s) to be submitted for background processing.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

operands optional, may include any desired TSO SUBMIT operands.

Remarks This interface is optional, it should only be used as an interface to the TSO Submit command.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display --------------  ROW 1 OF 6
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LIB.CNTL,VOL=SER=STR802  MEM=PDSET*  ---------------------------
>----->sub pdsback
>----->sub pdset*
PDS165I Members are: PDSETT1A, PDSETT1D, PDSETT1I, PDSETT1M, PDSETT1P,
         PDSETT1S

PDS395A Should these members be submitted (Yes/No) ?
>----->y
****************************** BOTTOM OF DATA  *******************************
```

Figure 150. Sample SUBMIT Subcommand

STATUS Function

Purpose The STATUS command displays the status of the ISPMODE functions associated with the CALC, CAX, CMDTBL, CSECTS, HEX, GO, LISTA (or DDNAME), LISTC (or LISTF), LISTV, LOG, MEMLIST, PBROWSE, PEDIT, WORKPAD and ZAP functions.

Example STATUS

Syntax

```
STATUS
```

Aliases ST, STA, STAT, STATU, STATUS

Operands (no operands are supported for the STATUS command).

Remarks

The STATUS display is provided to show the status of all current ISPMODE functions. This display is provided in response to a STATUS command or a STATUS response from the END confirmation panel.

In STARWARP each function is available concurrently. This means that as you create functions (log tables, MEMLIST tables, LISTA/DDNAME tables, LISTC/LISTF tables, and so forth) you may transit from one to another without losing the current function. When you select a function, you are placed back in it in an intelligent fashion. For example in the log, you are placed at the top of the log for the last command; however, if you had previously repositioned the log and you select it again with no log output additions, you will be placed at the same point again. Similarly, with the table panels, you will normally be positioned where you left the table unless one or more line commands has acted on the table; in that case, positioning would be for the last selected item.

A STARWARP function remains ACTIVE because the display table and related status information is retained even if you transit to a different table. Functions that have never been selected or have been deleted are marked INACTIVE. You can activate such functions by selecting them and supplying any information required for initialization.

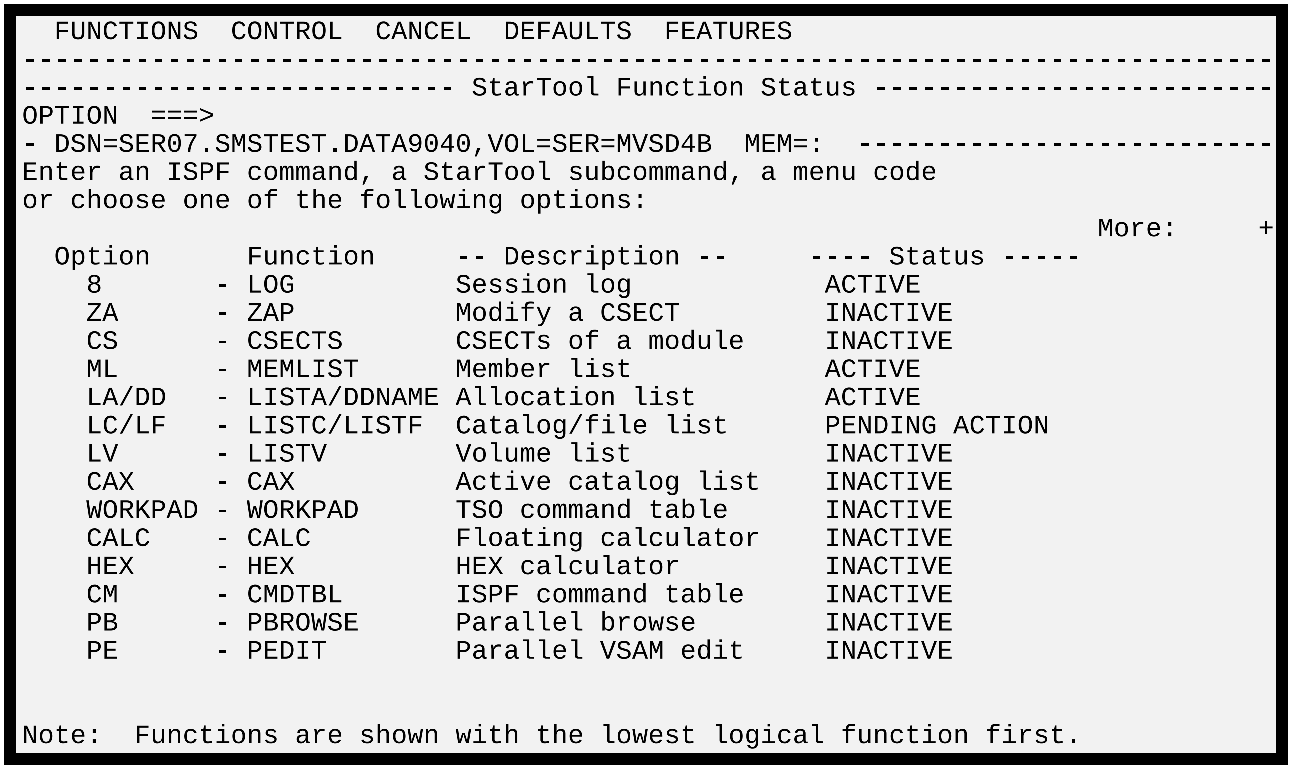
The final status possible is PENDING. PENDING indicates that the associated function has one or more incomplete line commands. If you select the function by entering its name, you direct the first pending line command to complete. Also, the STATUS function allows you to cancel pending line commands by function.

As noted above, one method of continuing a PENDING line command is to explicitly select the function by name. This will allow the current line command to complete and will initiate the next line command in that function. If no other line commands are pending, you will transit to the display for the selected function.

Another way to select PENDING line commands for processing is the **END** command. When an END command is issued, pending functions are selected for processing using a dynamic hierarchical structure in the order shown in the STATUS function display: LOG, ZAP, CSECTS, MEMLIST, LISTA (or DDNAME), LISTC (or LISTF), LISTV, CAX, WORKPAD, CALC, HEX, CMDTBL, PBROWSE and PEDIT. If there are no pending functions and the END was issued from one of the table displays noted above, the END command will position you to the log screen. If the END was issued from a log screen and no line commands are pending, STARWARP treats the END as a request to exit the program. By default, STARWARP will display an END Confirmation panel and continue or terminate based on your response to the END Confirmation panel (this default can be changed in SETALL).

A final way to select PENDING line commands for processing is the **PEND** command. When an PEND command is issued, pending functions are selected for processing using a dynamic hierarchical structure in the order shown in the STATUS function display below. If there are no pending functions, PEND has no effect.

Line commands in one function can generate processing for another function. STARWARP will dynamically create a hierarchical relation between the two functions. This is best illustrated by an ATTRIB line command from a MEMLIST screen whose line command output is generated in the log. You are automatically placed in the log screen to review the output; however, you may not enter additional MEMLIST commands (other than MEMLIST with no operands) until the pending line command is completed. If you enter MEMLIST or ML with no operands, MEMLIST completes its line command processing and dynamically removes the hierarchical relationship with the log.



```
  FUNCTIONS  CONTROL  CANCEL  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- StarTool Function Status -------------------------
OPTION  ===>
- DSN=SER07.SMSTEST.DATA9040,VOL=SER=MVSD4B  MEM=:  --------------------------
Enter an ISPF command, a StarTool subcommand, a menu code
or choose one of the following options:
                                                                   More:     +
  Option      Function     -- Description --     ---- Status -----
    8       - LOG          Session log            ACTIVE
    ZA      - ZAP          Modify a CSECT         INACTIVE
    CS      - CSECTS       CSECTs of a module     INACTIVE
    ML      - MEMLIST      Member list            ACTIVE
    LA/DD   - LISTA/DDNAME Allocation list        ACTIVE
    LC/LF   - LISTC/LISTF  Catalog/file list      PENDING ACTION
    LV      - LISTV        Volume list            INACTIVE
    CAX     - CAX          Active catalog list    INACTIVE
    WORKPAD - WORKPAD      TSO command table      INACTIVE
    CALC    - CALC         Floating calculator    INACTIVE
    HEX     - HEX          HEX calculator         INACTIVE
    CM      - CMDTBL       ISPF command table     INACTIVE
    PB      - PBROWSE      Parallel browse        INACTIVE
    PE      - PEDIT        Parallel VSAM edit     INACTIVE


Note:  Functions are shown with the lowest logical function first.
```

Figure 151. Sample STATUS Display

SVCMAP Subcommand

Purpose The SVCMAP subcommand formats information regarding the SVC table. SVC's are extensions to the operating system which can be invoked to perform system tasks. SVC's receive control in supervisor state and in the master storage protect key (key zero). They have the power to alter otherwise protected storage areas or issue privileged instructions that cannot be issued directly from problem programs.

If no SVC number is requested, a formatted list of all 256 SVC's is provided. If a SVC number is specified, only that SVC is formatted. If the SVC is an ESR (router) the ESR entries will all be formatted unless a particular ESR entry is requested.

Type 1 SVC's cannot link to other routines and cannot issue other SVC's from within the routine. They receive control with the local lock held and can obtain all other locks. They reside in the system nucleus (IEANUC0x).

Type 2 SVC's can link to other routines and can issue other SVC's themselves. They require no locks but can issue them. Like type 1 SVC's, they reside in the system nucleus.

Type 3/4 (3 or 4) SVC's can link to other routines and can issue other SVC's. They require no locks but can issue them. They reside in the Link Pack Area (LPA) or Modified Link Pack Area (MLPA). They can be loaded from any of the LPA libraries or alternatively, from SYS1.SVCLIB or SYS1.LINKLIB.

Type 6 SVC's cannot link to other routines or issue other SVC's. They execute disabled, meaning that no other program can gain control within MVS during the execution of the SVC. Locks cannot be held on entry to the SVC. Like types 1 and 2, type 6 SVC's reside in the system nucleus.

Four SVC's are ESR Router SVC's. SVC 109 contains a SVCTABLE for type 3 or 4 ESR SVC's, SVC 116 contains a SVCTABLE for type 1 ESR SVC's, SVC 122 contains a SVCTABLE for type 2 ESR SVC's, and SVC 137 contains a SVCTABLE for type 6 ESR SVC's.

ESR SVC's are identical in format to the SVC table except for the first eight bytes. The first three bytes contains "ESR" which is followed by a literal 1, 2, 3 or 6 depending on which SVC type it contains. The second four bytes contain a binary number indicating how many ESR SVC's it contains.

If the module name output by SVCMAP is ???, this indicates that the module could not be located in the nucleus, LPA or MLPA. This is usually because the SVC was dynamically added to the SVC table by an authorized program after the system IPL. SVCMAP will dump only the DEFAULT length of these routines since the length is unknown too.

Example

```
SVC   109  ESR(11)  DISASM
```

SVCMAP Subcommand

Syntax

```
SVCMAP   [number]  [ESR(entry)    ]
                   [DUMP          ]
                   [DISASM        ]
                   [CHANGE        ]
                   [DEFAULT(size) ]
                   [MAXLEN(size)  ]
```

Aliases SV, SVC, SVCM, SVCMA, SVCMAP

Defaults DEFAULT(256)

Required none

Operands

number optional, specifies the SVC to be investigated. If no number is specified, all 256 SVC's are listed.

ESR(entry) specifies the ESR entry number to be investigated. If this operand is left off on an ESR SVC, all ESR entries for the SVC will be listed. Note that this operand only has meaning on an ESR SVC.

DUMP specifies that the SVC module should be dumped. If the module can not be located in the nucleus, LPA or MLPA (probably dynamically loaded), only the DEFAULT length of the module is to be dumped.

The dump begins at the offset of the module indicated by the address from the SVCTABLE entry and the length is calculated to be the rest of the module regardless of other entry points.

The format of the dump is six or eight bytes of storage address, six bytes of hexadecimal offset into the module, followed by sixteen hexadecimal bytes of the module at that offset, and the character equivalent surrounded by asterisks.

DISASM specifies that the SVC module should be disassembled. If the module can not be located in the nucleus, LPA or MLPA (probably dynamically loaded), only the DEFAULT length of the module is to be formatted.

The display begins at the offset of the module indicated by the address from the SVCTABLE entry and the length is calculated to be the rest of the module regardless of other entry points.

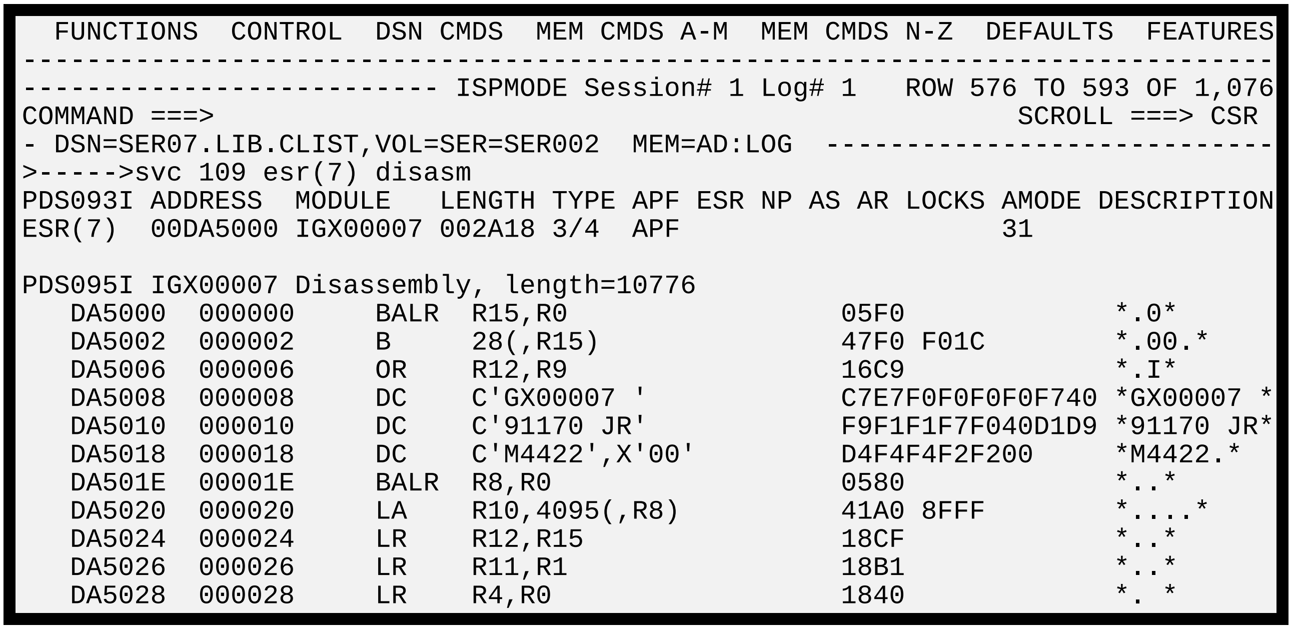
The format of the display is six or eight bytes of storage address, six bytes of hexadecimal offset into the module, followed by the operation code, reconstructed operands, the hexadecimal bytes of the module at that offset, and the character equivalent surrounded by asterisks.

CHANGE specifies that the FINDMOD routine should be invoked to attempt to locate the SVC module in the nucleus, linklist or LPALIB concatenation and if found, CHANGE to that data set.

DEFAULT(Sz) specifies the length to attempt to format with DUMP or DISASM for a module that is not found in the in-storage nucleus, LPA or MLPA. Sz may be coded as a number between 8 and 99999.

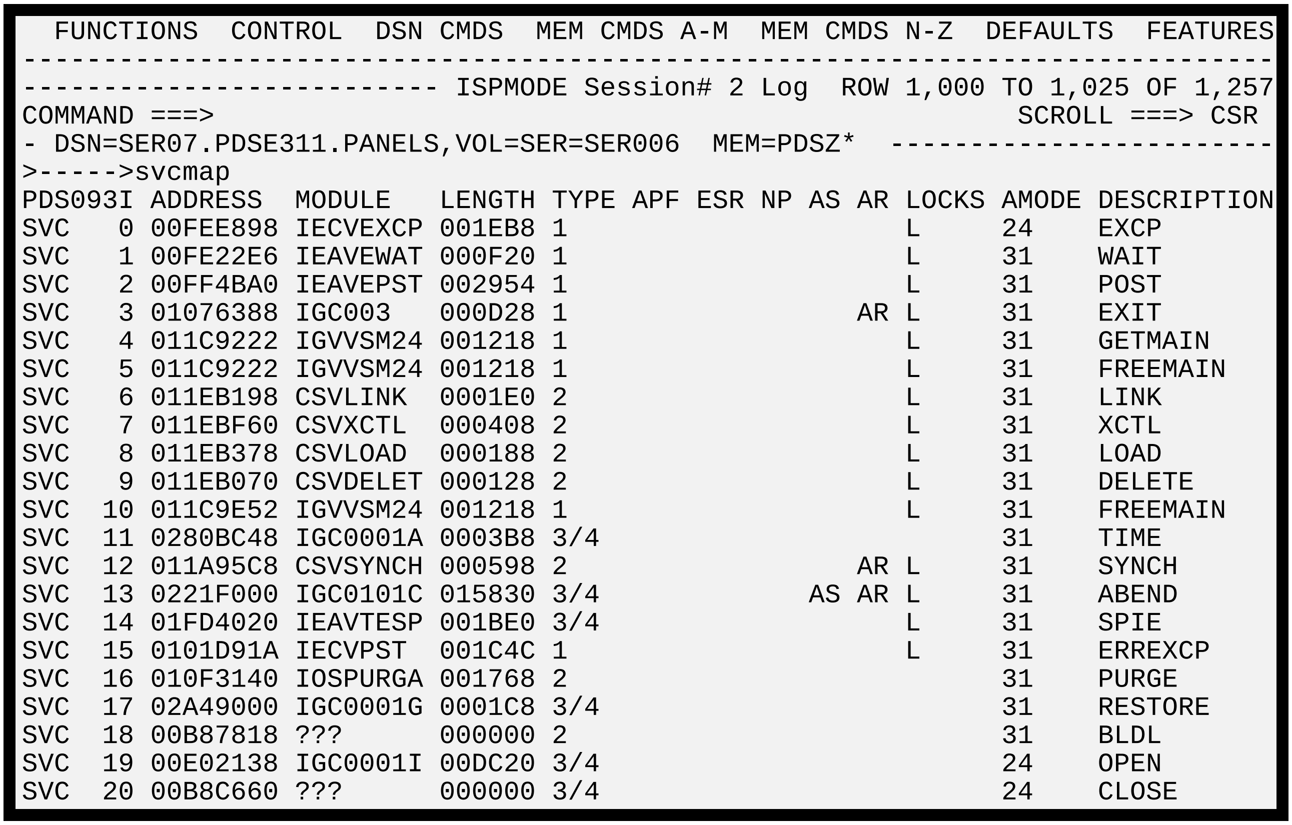
MAXLEN(Mx) specifies the maximum length to format with DUMP or DISASM. Mx may be coded as a number between 8 and 99999.

SVCMAP Subcommand



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log# 1   ROW 576 TO 593 OF 1,076
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CLIST,VOL=SER=SER002  MEM=AD:LOG  ----------------------------
>----->svc 109 esr(7) disasm
PDS093I ADDRESS  MODULE   LENGTH TYPE APF ESR NP AS AR LOCKS AMODE DESCRIPTION
ESR(7)  00DA5000 IGX00007 002A18 3/4  APF                    31

PDS095I IGX00007 Disassembly, length=10776
   DA5000  000000     BALR  R15,R0                 05F0             *.0*
   DA5002  000002     B     28(,R15)               47F0 F01C        *.00.*
   DA5006  000006     OR    R12,R9                 16C9             *.I*
   DA5008  000008     DC    C'GX00007 '            C7E7F0F0F0F0F740 *GX00007 *
   DA5010  000010     DC    C'91170 JR'            F9F1F1F7F040D1D9 *91170 JR*
   DA5018  000018     DC    C'M4422',X'00'         D4F4F4F2F200     *M4422.*
   DA501E  00001E     BALR  R8,R0                  0580             *..*
   DA5020  000020     LA    R10,4095(,R8)          41A0 8FFF        *....*
   DA5024  000024     LR    R12,R15                18CF             *..*
   DA5026  000026     LR    R11,R1                 18B1             *..*
   DA5028  000028     LR    R4,R0                  1840             *. *
```

Figure 152. Sample SVCMAP Subcommand with DISASM



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 2 Log  ROW 1,000 TO 1,025 OF 1,257
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.PDSE311.PANELS,VOL=SER=SER006  MEM=PDSZ*  ------------------------
>----->svcmap
PDS093I ADDRESS  MODULE   LENGTH TYPE APF ESR NP AS AR LOCKS AMODE DESCRIPTION
SVC   0 00FEE898 IECVEXCP 001EB8 1                     L     24    EXCP
SVC   1 00FE22E6 IEAVEWAT 000F20 1                     L     31    WAIT
SVC   2 00FF4BA0 IEAVEPST 002954 1                     L     31    POST
SVC   3 01076388 IGC003   000D28 1                  AR L     31    EXIT
SVC   4 011C9222 IGVVSM24 001218 1                     L     31    GETMAIN
SVC   5 011C9222 IGVVSM24 001218 1                     L     31    FREEMAIN
SVC   6 011EB198 CSVLINK  0001E0 2                     L     31    LINK
SVC   7 011EBF60 CSVXCTL  000408 2                     L     31    XCTL
SVC   8 011EB378 CSVLOAD  000188 2                     L     31    LOAD
SVC   9 011EB070 CSVDELET 000128 2                     L     31    DELETE
SVC  10 011C9E52 IGVVSM24 001218 1                     L     31    FREEMAIN
SVC  11 0280BC48 IGC0001A 0003B8 3/4                         31    TIME
SVC  12 011A95C8 CSVSYNCH 000598 2                  AR L     31    SYNCH
SVC  13 0221F000 IGC0101C 015830 3/4             AS AR L     31    ABEND
SVC  14 01FD4020 IEAVTESP 001BE0 3/4                   L     31    SPIE
SVC  15 0101D91A IECVPST  001C4C 1                     L     31    ERREXCP
SVC  16 010F3140 IOSPURGA 001768 2                           31    PURGE
SVC  17 02A49000 IGC0001G 0001C8 3/4                         31    RESTORE
SVC  18 00B87818 ???      000000 2                           31    BLDL
SVC  19 00E02138 IGC0001I 00DC20 3/4                         24    OPEN
SVC  20 00B8C660 ???      000000 3/4                         24    CLOSE
```

Figure 153. Sample SVCMAP Subcommand

TSO Subcommand

Purpose The TSO subcommand may be used to invoke a TSO command processor or CLIST. Any desired operands may be specified after the name specified.

Example TSO listc lev(sys2)

Syntax

```
TSO   command [operands]
```

Aliases T, TS, TSO

Defaults none

Required command

Operands

command identifies the CLIST or command processor to invoke. If the form %cnam if used, CNAM is assumed to be a CLIST name.

operands optional, may include any desired operands.

Remarks The TSO subcommand may be used to invoke a TSO command processor or CLIST. Any desired operands may be specified after the name specified.

When in ISPMODE or MEMLIST, if **TSO** is entered, this is the ISPF TSO command. If you wish to use the STARWARP TSO subcommand, you must enter an alias subcommand name such as T or TS.

Your installation may have installed the TSO subcommand to obtain the equivalent of "TSO TSOEXEC command operands". If this is the case, you can invoke authorized commands with a few less keystrokes. To determine if your installation has this option enabled, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "TSO calls"; A value of TSOEXEC indicates it is enabled.

TSOEDIT Subcommand

Purpose The TSOEDIT subcommand edits a member; The TSO EDIT command is used.

Example TSOEDIT mema:memb cntl

Syntax

```
TSOEDIT   memgroup [ASIS        ]
                   [OLD/NEW     ]
                   [NONUM       ]
                   [ASM/BASIC/CLIST/CNTL/COBOL/DATA/
                    FORTE/FORTG/FORTGI/FORTH/GOFORT/
                    IPLI/LIST/PLI/PLIF/TEXT/VSBASIC ]
```

Aliases TSOE, TSOED, TSOEDI, TSOEDIT

Defaults memgroup, EDIT type based on the data set name

Required none

Operands

memgroup identifies the member(s) to be edited.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

ASIS	edit the member with upper and lower case characters.
OLD	verify that the member exists before calling EDIT.
NEW	verify that the member does not exist before calling EDIT.
NONUM	edit the member without using line numbers.
ASM	EDIT type is assembly
BASIC	EDIT type is BASIC
CLIST	EDIT type is CLIST
CNTL	EDIT type is CNTL
COBOL	EDIT type is COBOL
DATA	EDIT type is DATA
FORTE	EDIT type is FORTRAN level "E"
FORTG	EDIT type is FORTRAN level "G"
FORTGI	EDIT type is FORTRAN level "GI"
FORTH	EDIT type is FORTRAN level "H"
GOFORT	EDIT type is GOFORT
IPLI	EDIT type is IPLI
LIST	EDIT type is LIST
PLI	EDIT type is PLI
PLIF	EDIT type is PLI (F-level)
TEXT	EDIT type is TEXT
VSBASIC	EDIT type is VSBASIC

Remarks If an EDIT type keyword is entered, that keyword is passed to EDIT as the descriptive qualifier; otherwise, a descriptive qualifier is chosen as follows:

1. If the low-level qualifier of the partitioned data set name is one of the valid descriptive qualifiers for EDIT (ASM, BASIC, CLIST, CNTL, COBOL, DATA, IPLI, LIST, PLI, TEXT or VSBASIC), that qualifier is passed to EDIT.
2. If the low-level qualifier is FORT, GOFORT is passed to EDIT. Otherwise, the data type qualifier is not a valid descriptive qualifier and the general descriptive qualifier, DATA, is passed to EDIT.

This interface is optional, it should only be used as an interface to the TSO EDIT command. To terminate this interface, enter END from the EDIT prompt.

TSOLIST Subcommand

This interface is optional, it should only be used if your installation has installed the public domain LIST command.

Purpose The TSOLIST subcommand lists a member. The TSO LIST command is used; any desired LIST operands may be specified after the member name.

Example TSOLIST   mema:memb

Syntax

```
TSOLIST   memgroup  [BYPASS(numtoskip)    ]
                    [COL(begincol:endcol) ]
                    [HEX/X                ]
                    [MAX(maxtolist)       ]
                    [ NUM/SNUM/NONUM       ]
```

Aliases TSOL, TSOLI, TSOLIS, TSOLIST

Defaults memgroup, NUM (if a non-numeric is encountered, then NONUM)

Required none

Operands

memgroup identifies the member(s) to be listed.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

others optional, may include any desired TSO LIST operands.

USAGE Subcommand

Purpose The USAGE subcommand displays statistics on the current data set. Note that the DSNAME subcommand is automatically issued with this subcommand.

Example USAGE

Syntax

```
USAGE   [ALL]
```

Aliases U, US, USA, USAG, USAGE

Defaults none

Required none

Operands

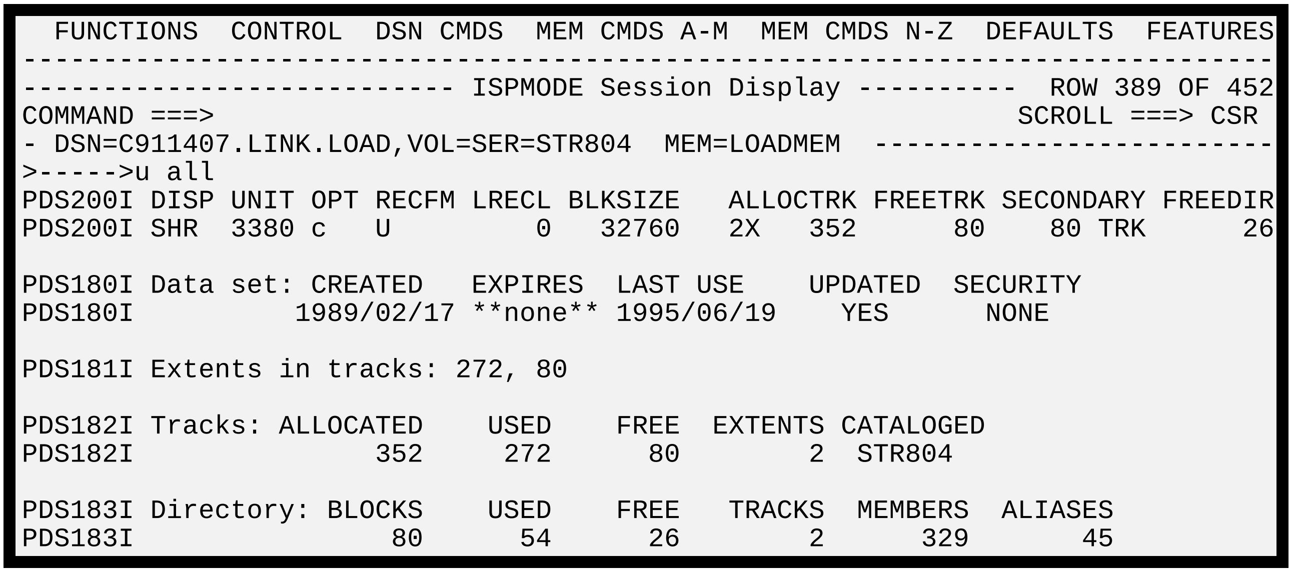
ALL display data set extent information and format the DSCB fields. Actually, any operand on the USAGE subcommand is sufficient to get all of the available data.

Remarks The statistics displayed include:

- the number of disk storage tracks allocated the number of disk storage tracks currently in use and free
- the number of disk storage extents required for the data set
- the number of directory blocks allocated
- the number of directory blocks currently in use and free
- the number of disk storage tracks used by the directory
- the number of members in the directory
- the number of member names which are alias names

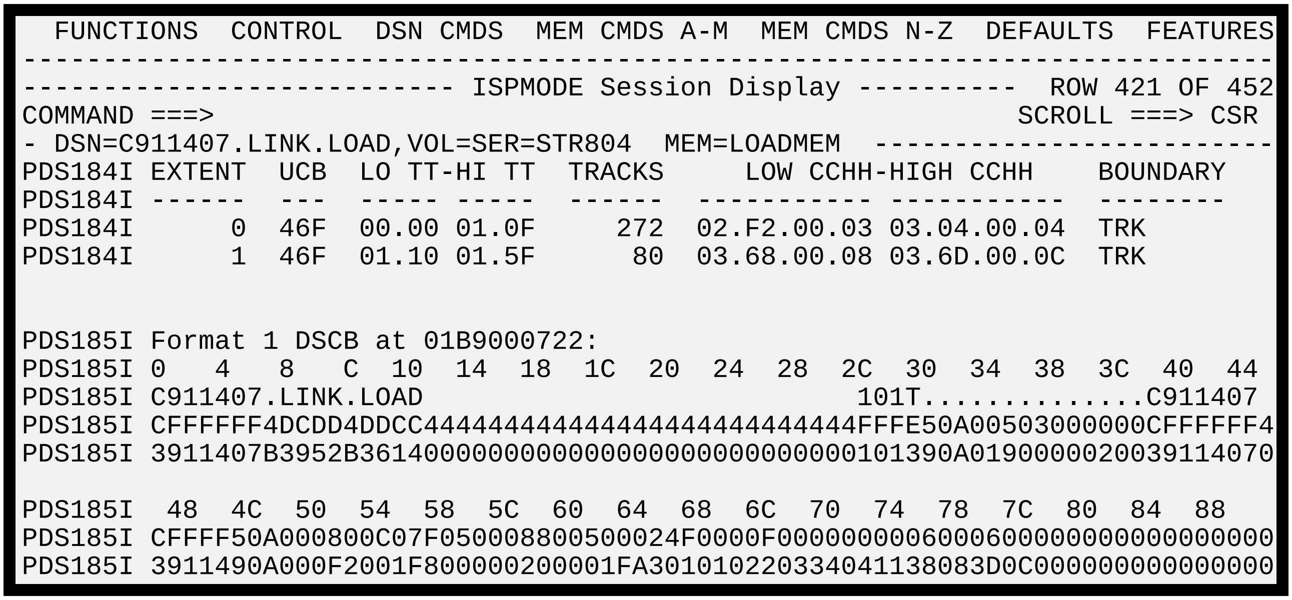
USAGE Subcommand

Following are some sample screens of the USAGE subcommand. Note that USAGE with no operand would return data from the first screen and the remaining screens would be added for any operand after USAGE.



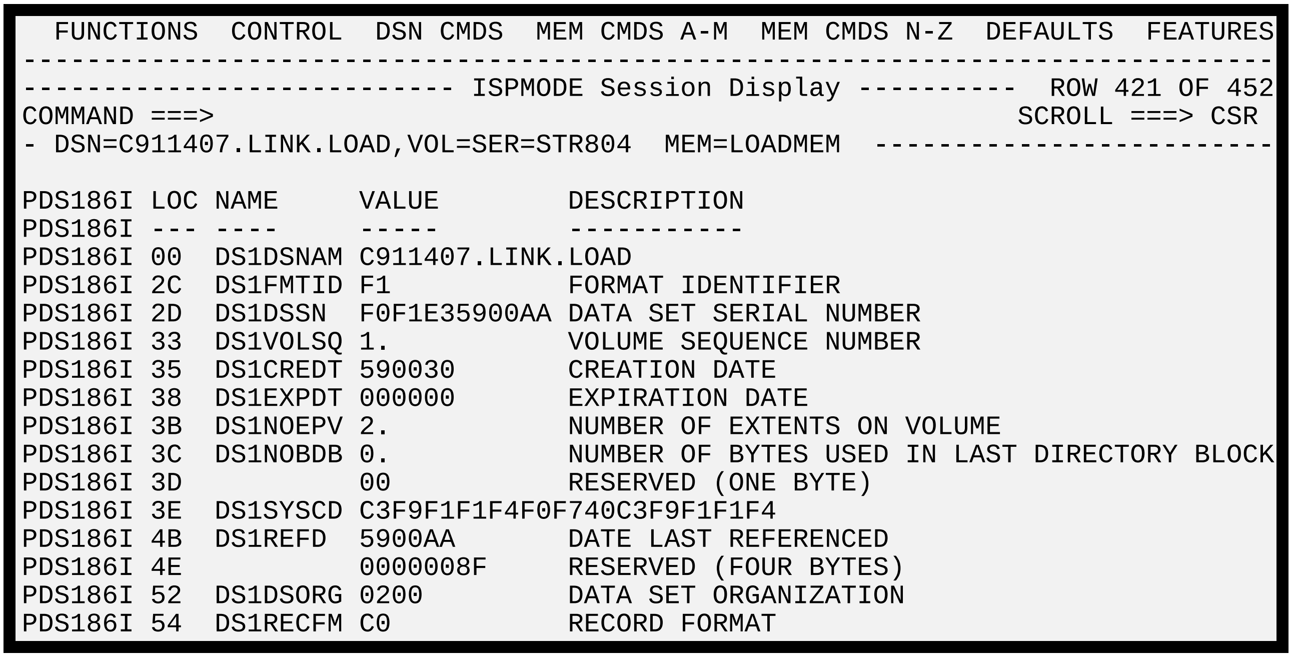
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 389 OF 452
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM  ------------------------
>----->u all
PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY FREEDIR
PDS200I SHR  3380 c   U         0   32760   2X   352      80    80 TRK      26

PDS180I Data set: CREATED   EXPIRES  LAST USE    UPDATED  SECURITY
PDS180I          1989/02/17 **none** 1995/06/19    YES      NONE

PDS181I Extents in tracks: 272, 80

PDS182I Tracks: ALLOCATED    USED    FREE  EXTENTS CATALOGED
PDS182I               352     272      80        2  STR804

PDS183I Directory: BLOCKS    USED    FREE   TRACKS  MEMBERS  ALIASES
PDS183I                80      54      26        2      329       45
```

Figure 154. Sample USAGE part 1



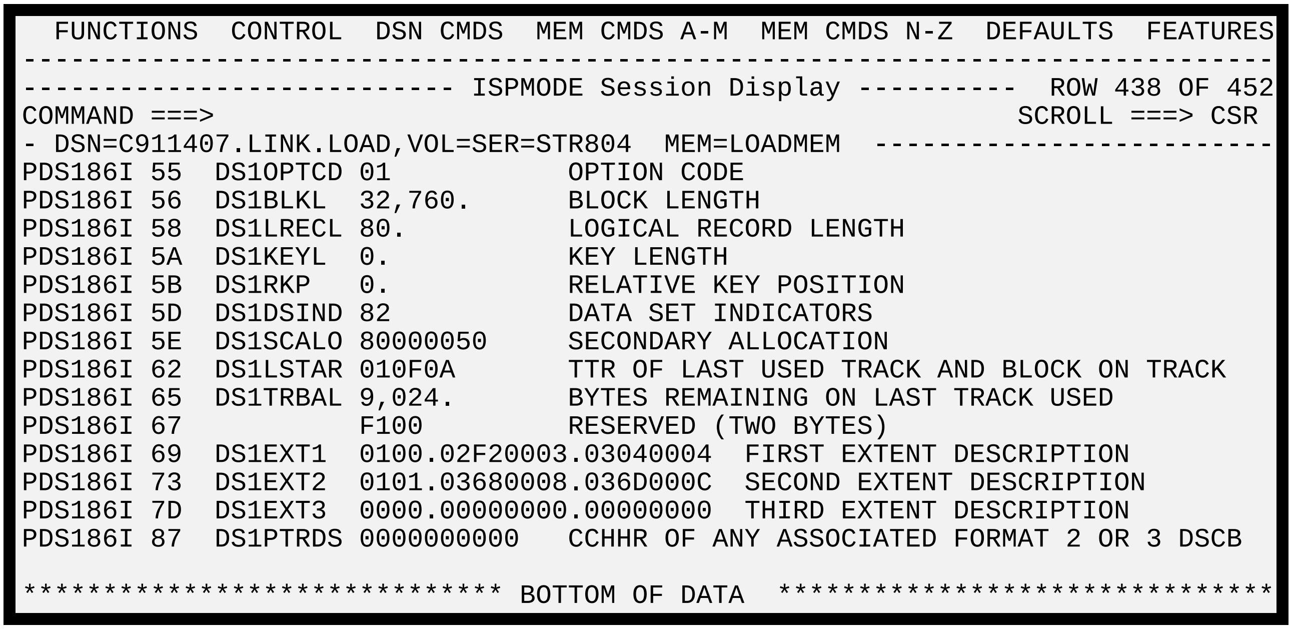
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session Display ----------  ROW 421 OF 452
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM  ------------------------
PDS184I EXTENT  UCB  LO TT-HI TT  TRACKS     LOW CCHH-HIGH CCHH    BOUNDARY
PDS184I ------  ---  ----- -----  ------  ----------- -----------  --------
PDS184I      0  46F  00.00 01.0F     272  02.F2.00.03 03.04.00.04  TRK
PDS184I      1  46F  01.10 01.5F      80  03.68.00.08 03.6D.00.0C  TRK


PDS185I Format 1 DSCB at 01B9000722:
PDS185I 0   4   8   C  10  14  18  1C  20  24  28  2C  30  34  38  3C  40  44
PDS185I C911407.LINK.LOAD                           101T..............C911407
PDS185I CFFFFFF4DCDD4DDCC444444444444444444444444444FFFE50A00503000000CFFFFFF4
PDS185I 3911407B3952B3614000000000000000000000000000101390A01900000200391140 70

PDS185I  48  4C  50  54  58  5C  60  64  68  6C  70  74  78  7C  80  84  88
PDS185I CFFFF50A000800C07F050008800500024F0000F00000000060006000000000000000000
PDS185I 3911490A000F2001F800000200001FA30101022033404113808 3D0C000000000000000
```

Figure 155. Sample USAGE part 2



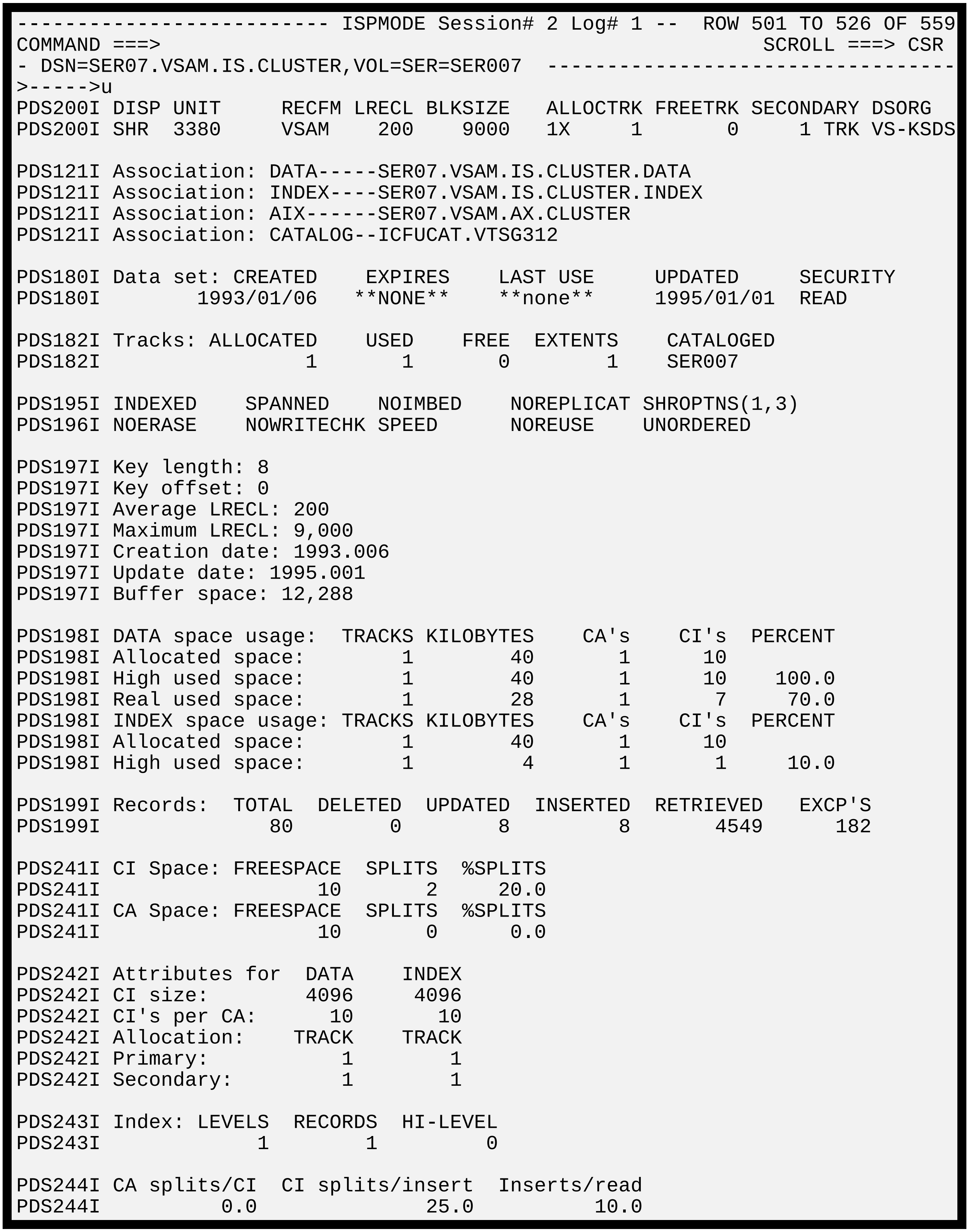
```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 421 OF 452
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM  -------------------------

PDS186I LOC NAME     VALUE        DESCRIPTION
PDS186I --- ----     -----        -----------
PDS186I 00  DS1DSNAM C911407.LINK.LOAD
PDS186I 2C  DS1FMTID F1           FORMAT IDENTIFIER
PDS186I 2D  DS1DSSN  F0F1E35900AA DATA SET SERIAL NUMBER
PDS186I 33  DS1VOLSQ 1.           VOLUME SEQUENCE NUMBER
PDS186I 35  DS1CREDT 590030       CREATION DATE
PDS186I 38  DS1EXPDT 000000       EXPIRATION DATE
PDS186I 3B  DS1NOEPV 2.           NUMBER OF EXTENTS ON VOLUME
PDS186I 3C  DS1NOBDB 0.           NUMBER OF BYTES USED IN LAST DIRECTORY BLOCK
PDS186I 3D           00           RESERVED (ONE BYTE)
PDS186I 3E  DS1SYSCD C3F9F1F1F4F0F740C3F9F1F1F4
PDS186I 4B  DS1REFD  5900AA       DATE LAST REFERENCED
PDS186I 4E           0000008F     RESERVED (FOUR BYTES)
PDS186I 52  DS1DSORG 0200         DATA SET ORGANIZATION
PDS186I 54  DS1RECFM C0           RECORD FORMAT
```

Figure 156. Sample USAGE part 3



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ----------  ROW 438 OF 452
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.LINK.LOAD,VOL=SER=STR804  MEM=LOADMEM  -------------------------
PDS186I 55  DS1OPTCD 01           OPTION CODE
PDS186I 56  DS1BLKL  32,760.      BLOCK LENGTH
PDS186I 58  DS1LRECL 80.          LOGICAL RECORD LENGTH
PDS186I 5A  DS1KEYL  0.           KEY LENGTH
PDS186I 5B  DS1RKP   0.           RELATIVE KEY POSITION
PDS186I 5D  DS1DSIND 82           DATA SET INDICATORS
PDS186I 5E  DS1SCALO 80000050     SECONDARY ALLOCATION
PDS186I 62  DS1LSTAR 010F0A       TTR OF LAST USED TRACK AND BLOCK ON TRACK
PDS186I 65  DS1TRBAL 9,024.       BYTES REMAINING ON LAST TRACK USED
PDS186I 67           F100         RESERVED (TWO BYTES)
PDS186I 69  DS1EXT1  0100.02F20003.03040004  FIRST EXTENT DESCRIPTION
PDS186I 73  DS1EXT2  0101.03680008.036D000C  SECOND EXTENT DESCRIPTION
PDS186I 7D  DS1EXT3  0000.00000000.00000000  THIRD EXTENT DESCRIPTION
PDS186I 87  DS1PTRDS 0000000000   CCHHR OF ANY ASSOCIATED FORMAT 2 OR 3 DSCB

****************************** BOTTOM OF DATA  *******************************
```

Figure 157. Sample USAGE part 4



```
------------------------- ISPMODE Session# 2 Log# 1 --  ROW 501 TO 526 OF 559
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.VSAM.IS.CLUSTER,VOL=SER=SER007  ---------------------------------
>----->u
PDS200I DISP UNIT     RECFM LRECL BLKSIZE   ALLOCTRK FREETRK SECONDARY DSORG
PDS200I SHR  3380     VSAM    200    9000   1X     1       0     1 TRK VS-KSDS

PDS121I Association: DATA-----SER07.VSAM.IS.CLUSTER.DATA
PDS121I Association: INDEX----SER07.VSAM.IS.CLUSTER.INDEX
PDS121I Association: AIX------SER07.VSAM.AX.CLUSTER
PDS121I Association: CATALOG--ICFUCAT.VTSG312

PDS180I Data set: CREATED    EXPIRES    LAST USE     UPDATED     SECURITY
PDS180I        1993/01/06   **NONE**    **none**     1995/01/01  READ

PDS182I Tracks: ALLOCATED    USED    FREE  EXTENTS    CATALOGED
PDS182I                 1       1       0        1    SER007

PDS195I INDEXED    SPANNED    NOIMBED    NOREPLICAT SHROPTNS(1,3)
PDS196I NOERASE    NOWRITECHK SPEED      NOREUSE    UNORDERED

PDS197I Key length: 8
PDS197I Key offset: 0
PDS197I Average LRECL: 200
PDS197I Maximum LRECL: 9,000
PDS197I Creation date: 1993.006
PDS197I Update date: 1995.001
PDS197I Buffer space: 12,288

PDS198I DATA space usage:  TRACKS KILOBYTES    CA's    CI's  PERCENT
PDS198I Allocated space:        1        40       1      10
PDS198I High used space:        1        40       1      10    100.0
PDS198I Real used space:        1        28       1       7     70.0
PDS198I INDEX space usage: TRACKS KILOBYTES    CA's    CI's  PERCENT
PDS198I Allocated space:        1        40       1      10
PDS198I High used space:        1         4       1       1     10.0

PDS199I Records:  TOTAL  DELETED  UPDATED  INSERTED  RETRIEVED   EXCP'S
PDS199I              80        0        8         8       4549      182

PDS241I CI Space: FREESPACE  SPLITS  %SPLITS
PDS241I                  10       2     20.0
PDS241I CA Space: FREESPACE  SPLITS  %SPLITS
PDS241I                  10       0      0.0

PDS242I Attributes for  DATA    INDEX
PDS242I CI size:        4096     4096
PDS242I CI's per CA:      10       10
PDS242I Allocation:    TRACK    TRACK
PDS242I Primary:           1        1
PDS242I Secondary:         1        1

PDS243I Index: LEVELS  RECORDS  HI-LEVEL
PDS243I             1        1         0

PDS244I CA splits/CI  CI splits/insert  Inserts/read
PDS244I          0.0              25.0          10.0
```

Figure 158. Sample USAGE for VSAM

VERIFY Subcommand

Purpose The VERIFY subcommand validity checks data sets. For PDS or PDSE data sets, the entire data set may be checked by entering **:** in the member name position. An individual member or member group may be checked by entering its name in the member name position.

Example VERIFY mema:memb

Syntax

```
VERIFY   memgroup
         [LOAD/NOLOAD              ]
         [LKED/NOLKED              ]
         [MAXBLK(bsize)            ]
         [NAME/NONAME              ]
         [COUNT/NOCOUNT            ]
         [READ/NOREAD/INPUT/NOINPUT]
         [STATS/NOSTATS            ]
         [UPDATE/NOUPDATE          ]
         [MEMBERS / MEMLIST / ML / NEWML / SUBLIST       ]
       * [FROMKEY(key)/FROMADDRESS(add)/FROMNUMBER(num)  ]
       * [TOKEY(key)/TOADDRESS(add)/TONUMBER(num)        ]
       * [EXAMINE / NOEXAMINE / NOINDEXTEST / NODATATEST ]
*NOTE: Lines with an asterisk are supported for VSAM only.
```

Aliases V, VE, VER, VERI, VERIF, VERIFY

Abbreviations FKEY for FROMKEY, FADDR for FROMADDRESS, FNUM for FROMNUMBER, TKEY for TOKEY, TADDR for TOADDRESS and TNUM for TONUMBER.

Defaults memgroup, LOAD, LKED, NAME, READ, STATS, NOUPDATE, EXAMINE

Required none

Operands

memgroup identifies the member(s) to be validity checked. If **:** is entered in the member name position, the data set is validity checked; otherwise, only the specified members are checked individually.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

LOAD for load libraries, LOAD each member.
NOLOAD do not LOAD any members.
LKED for load libraries, check linkage editor attributes and linklist LLA member status.
NOLKED do not perform any linkage editor checks or linklist LLA member status.
MAXBLK(bsize) generate messages for members with blocks larger than the value **bsize**.
NAME check member names for validity. Valid member names are from one to eight characters long containing upper case alphameric characters and the first character may not be numeric.
NONAME do not check member names for validity.
COUNT output a short message giving input counts.
NOCOUNT output normal messages.

READ input each member (or the data set).

NOREAD do not input any data.

INPUT input each member (or the data set).

NOINPUT do not read any data.

STATS display normal completion statistical messages.

NOSTATS do not display any statistical messages. This would normally be used to check a member or a data set for unusual conditions.

UPDATE test member addition and deletion to determine if the PDS directory has any available space and if normal PDS operations are possible on the data set. For a VSAM data set, perform an IDCAMS VERIFY function.

NOUPDATE do not update the data set.

MEMBERS displays the names of members with warning or error messages without changing the current member group.

MEMLIST Same as **ML**. Specifies that any member with warning or error messages will be selected for MEMLIST display. These messages are in the range PDS400W through PDS999E. If no members are selected, a null sublist is the result.

ML Same as **MEMLIST**. Specifies that any member with warning or error messages will be selected for MEMLIST display. These messages are in the range PDS400W through PDS999E. If no members are selected, a null sublist is the result.

NEWML Same as **MEMLIST** and **ML** except that the current MEMLIST is reset.

SUBLIST Specifies that any member with warning or error messages will be selected for inclusion in a new sublist. These messages are in the range PDS400W through PDS999E. If no members are selected, a null sublist is the result.

FROMKEY(ky) for VSAM data sets only, **ky** is coded as the key of the first record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access begins at the first record whose key matches (or is greater than) the portion of the key specified. This parameter may be used with TOKEY and it can only be specified for an alternate index or a key-sequenced data set.

FROMADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the first record to be accessed. The RBA value must match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with TOADDRESS.

FROMADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

FROMNUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the first record to be accessed. This record must be present in the data set.

This parameter may be used with TONUMBER and it can only be specified for a variable or fixed relative record data set.

TOKEY(ky) for VSAM data sets only, **ky** is coded as the key of the last record to be accessed. This is a generic key and it may be coded as **x'hexkey'**; access ends after the first record whose key matches the portion of the key specified.

This parameter may be used with FROMKEY and it can only be specified for an alternate index or a key-sequenced data set.

TOADDRESS(ad) for VSAM data sets only, **ad** is coded as 0 through 99999999 to specify the relative byte address (RBA) of the last record to be accessed. The RBA value does need not to match the beginning of a logical record. If this parameter is specified for key-sequenced data, the records will be accessed in physical sequential order instead of key order. This parameter may be used with FROMADDRESS.

TOADDRESS(address)

- Can be coded only for key-sequenced, entry-sequenced or linear data sets or DATA or INDEX components.
- Can not be specified if the data set is being accessed through a path.
- Can not be specified for a key-sequenced data set with spanned records if any of the spanned records are to be accessed.

TONUMBER(nm) for VSAM data sets only, **nm** is coded as 0 through 99999999 to specify the relative record number (RRN) of the last record to be accessed. This record need not be present in the data set.

This parameter may be used with FROMNUMBER and it can only be specified for a variable or fixed relative record data set.

EXAMINE for VSAM data sets only, specifies that an IDCAMS EXAMINE with INDEXTEST and DATATEST are desired.

NOEXAMINE for VSAM data sets only, specifies that no IDCAMS EXAMINE is desired.

NOINDEXTEST for VSAM data sets only, specifies that an IDCAMS EXAMINE with NOINDEXTEST is desired.

NODATATEST for VSAM data sets only, specifies that an IDCAMS EXAMINE with NODATATEST is desired.

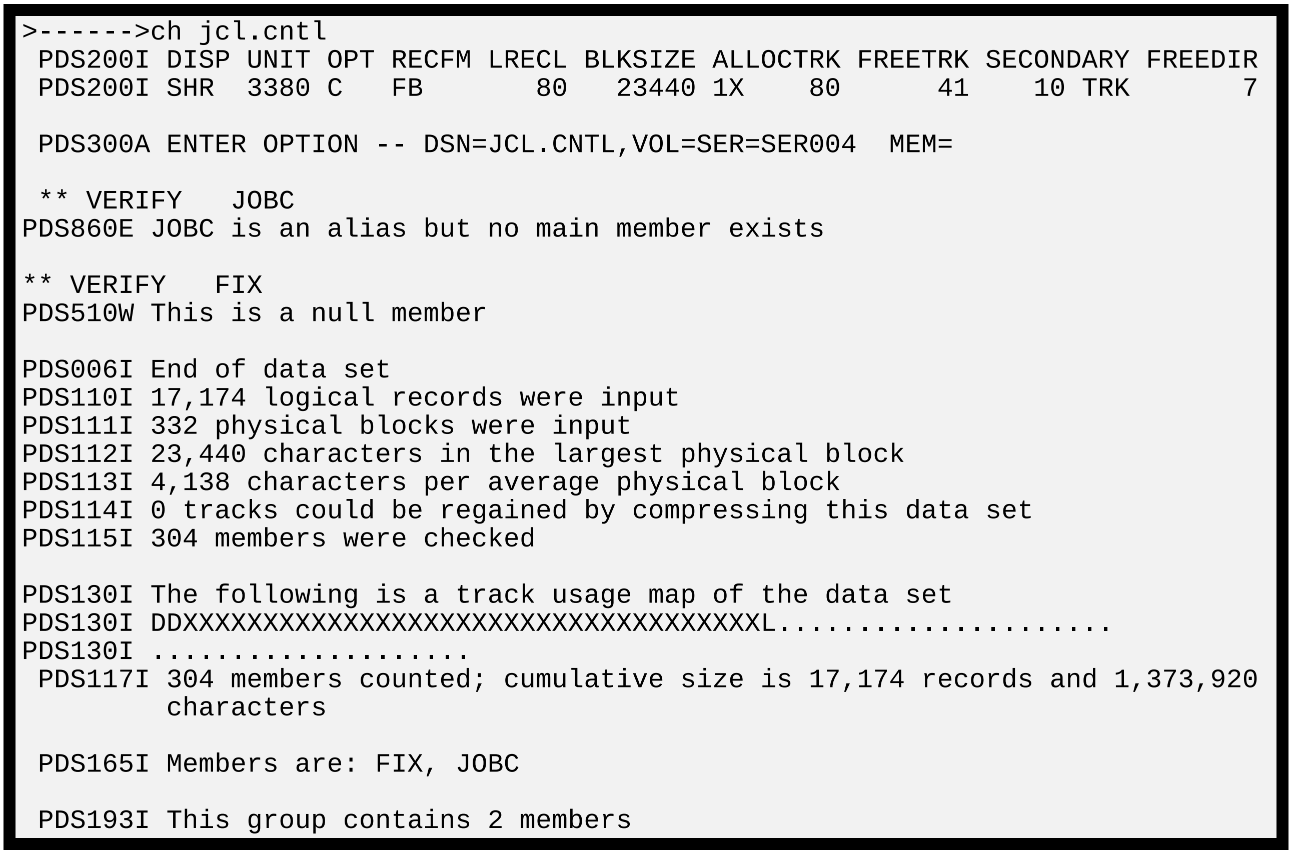
Remarks The VERIFY subcommand validity checks PDS or PDSE data sets in the following manner:

1. For **VERIFY : UPDATE**, a dummy member is added and deleted to determine if the data set directory is full and if normal PDS functions are possible. Note that this is the only potential data set update performed by the VERIFY subcommand and it is not performed on a PDSE data set.
2. The data set directory is read, member names are placed into a program table in TTR sort order (member location order) and the following checks are performed:
   a) If NAME is in effect, member names are checked for validity.
   b) If the data set is a linklist library and LKED is in effect, the LLA status of members is checked with a system BLDL.
      i. A message is issued if the LLA status of a member can not be determined due to a module in a TASKLIB library or a member in a higher linklist library.
      ii. A message is issued if a member's directory entry and its LLA entry are not identical (they are not synchronized).
      iii. A message is issued if a linklist member is not known to LLA.
   c) If the data set is a load library and LKED is in effect, the following illegal editor attribute combinations are checked:
      i. Modules with RENT and NOREUS attributes
      ii. Modules with RMODEANY and AMODE24 or AMODEANY attributes
      iii. Modules with OVLY and RENT, REUS, REFR, SCTR, RMODEANY, AMODE31, or AMODEANY attributes
      iv. Modules with TEST and NOEDIT attributes
      v. Modules with REUS and SCTR attributes
   d) If the data set is a load library and LOAD is in effect, each member is checked for ABENDS during a LOAD operation.
   e) Each member is checked for current usage by an ISPF EDIT session.
   f) Member directory entries are checked for correct name order.

   g) Each member's TTR address is checked against the end of the data set.
3. The program TTR table is checked for:
   a) Orphan members (aliases with no associated main member).
   b) Apparent alias members (main members whose start address matches another main member's start address).
   c) Alias load modules with incorrect directory pointers to associated main members.
   d) Alias load modules whose RMODE entries or main AMODE entries do not match their base module's entries.
4. If READ or INPUT is in effect, the directory and each member is read checking for:
   a) Permanent I/O errors
   b) Input blocksizes exceeding the MAXBLK value (or the DCB BLKSIZE if MAXBLK is omitted)
   c) BLKSIZE divided by input LRECL not integral (RECFM=F)
   d) Input LRECL exceeding maximum DCB LRECL (RECFM=V)
   e) Input LRECL less than 4 bytes (RECFM=V)
   f) Null members
   g) Directory RLD/CONTROL counts which do not match the first RLD entry
5. Informational messages regarding the data set are provided.

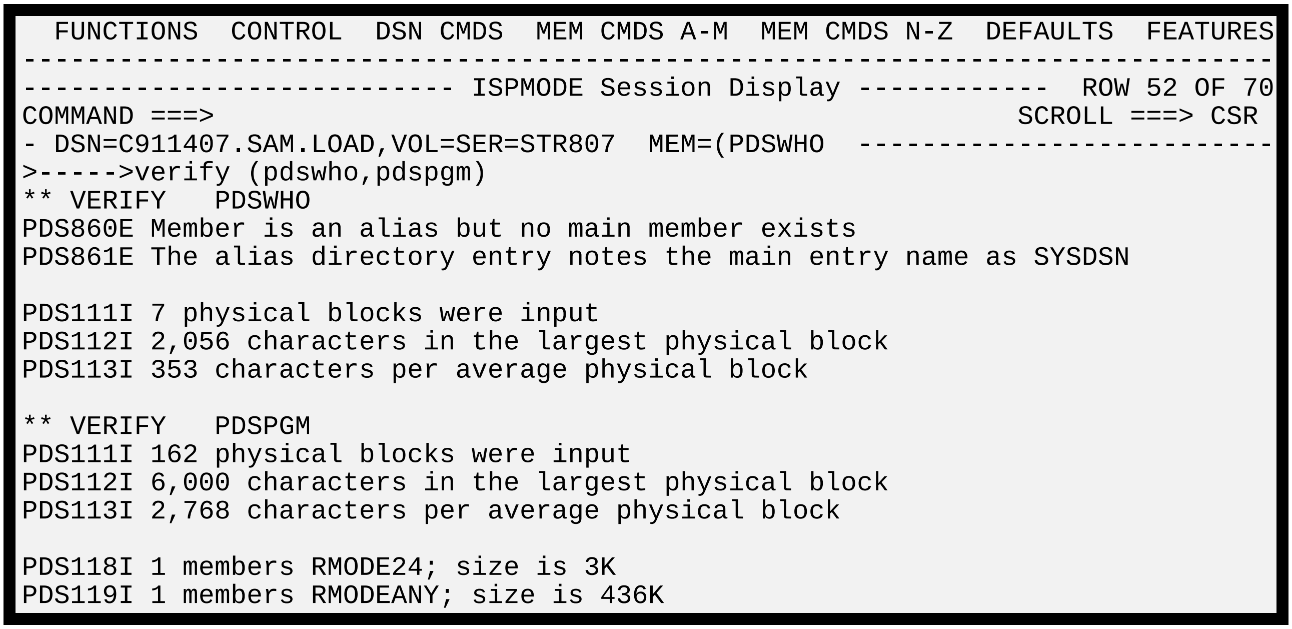
The VERIFY subcommand validity checks sequential or VSAM data sets in the following manner:

1. For VSAM data sets if NOUPDATE is not specified, an IDCAMS VERIFY is performed on the data set.
2. For a KSDS or an Alternate index if NOEXAMINE is not specified, an IDCAMS EXAMINE is performed on the data set.
3. For a KSDS or an Alternate index if NOREAD is not specified, the data set is also read sequentially to determine if the index is synchronized with the data.
4. Sequential data sets are checked for current usage by an ISPF EDIT session.
5. If READ or INPUT is in effect, the data set is read checking for:
   a) Permanent I/O errors or VSAM logical errors.
   b) Input blocksizes exceeding the MAXBLK value (or the DCB BLKSIZE if MAXBLK is omitted)
   c) BLKSIZE divided by input LRECL not integral (RECFM=F)
   d) Input LRECL exceeding maximum DCB LRECL (RECFM=V)
   e) Input LRECL less than 4 bytes (RECFM=V)
6. Informational messages regarding the data set are provided.



```
>------>ch jcl.cntl
 PDS200I DISP UNIT OPT RECFM LRECL BLKSIZE ALLOCTRK FREETRK SECONDARY FREEDIR
 PDS200I SHR  3380 C   FB       80   23440 1X    80      41    10 TRK       7

 PDS300A ENTER OPTION -- DSN=JCL.CNTL,VOL=SER=SER004  MEM=

 ** VERIFY   JOBC
PDS860E JOBC is an alias but no main member exists

** VERIFY   FIX
PDS510W This is a null member

PDS006I End of data set
PDS110I 17,174 logical records were input
PDS111I 332 physical blocks were input
PDS112I 23,440 characters in the largest physical block
PDS113I 4,138 characters per average physical block
PDS114I 0 tracks could be regained by compressing this data set
PDS115I 304 members were checked

PDS130I The following is a track usage map of the data set
PDS130I DDXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXL.....................
PDS130I ....................
 PDS117I 304 members counted; cumulative size is 17,174 records and 1,373,920
         characters

 PDS165I Members are: FIX, JOBC

 PDS193I This group contains 2 members
```

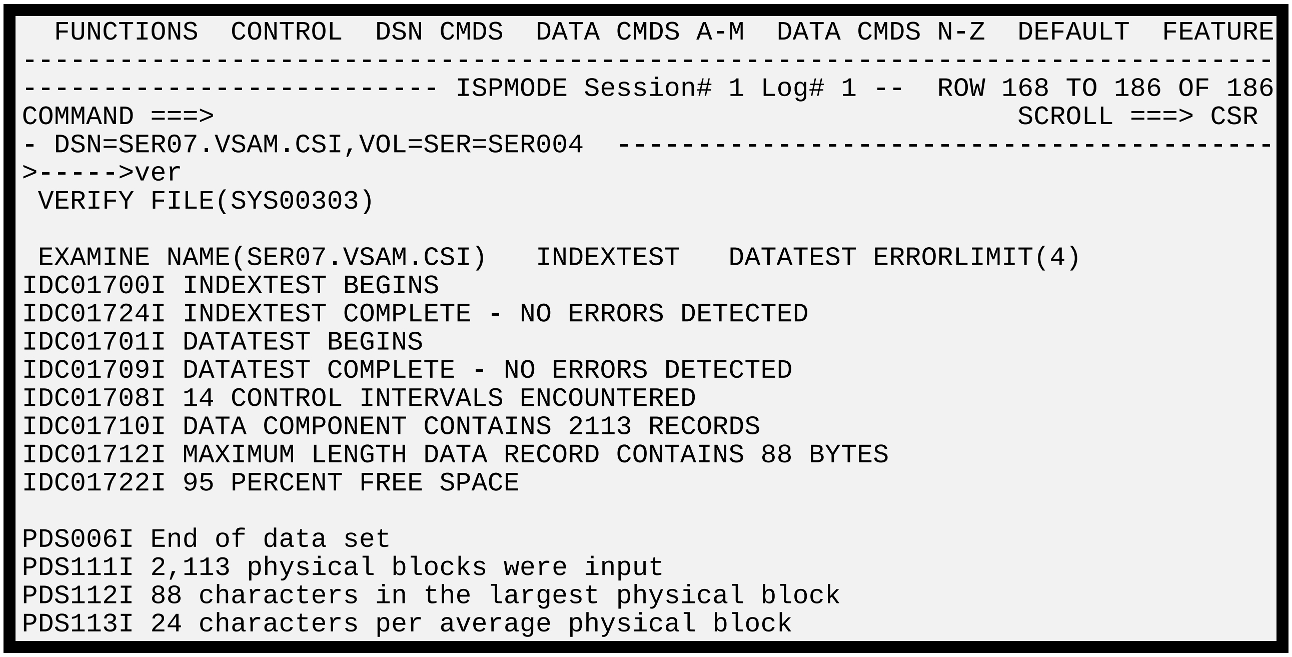
Figure 159. Sample VERIFY Subcommand (source)



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
--------------------------- ISPMODE Session Display ------------  ROW 52 OF 70
COMMAND ===>                                                 SCROLL ===> CSR
- DSN=C911407.SAM.LOAD,VOL=SER=STR807  MEM=(PDSWHO  --------------------------
>----->verify (pdswho,pdspgm)
** VERIFY   PDSWHO
PDS860E Member is an alias but no main member exists
PDS861E The alias directory entry notes the main entry name as SYSDSN

PDS111I 7 physical blocks were input
PDS112I 2,056 characters in the largest physical block
PDS113I 353 characters per average physical block

** VERIFY   PDSPGM
PDS111I 162 physical blocks were input
PDS112I 6,000 characters in the largest physical block
PDS113I 2,768 characters per average physical block

PDS118I 1 members RMODE24; size is 3K
PDS119I 1 members RMODEANY; size is 436K
```

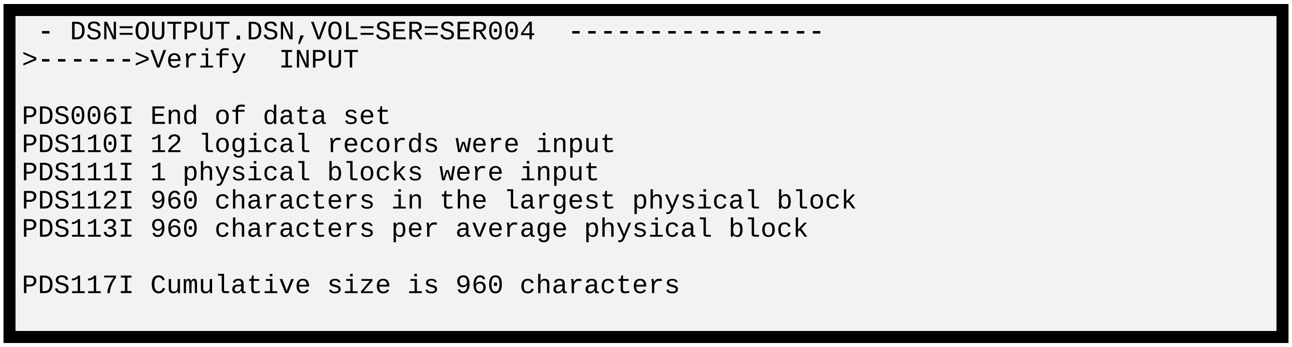
Figure 160. Sample VERIFY Subcommand (load)

VERIFY Subcommand



```
  FUNCTIONS  CONTROL  DSN CMDS  DATA CMDS A-M  DATA CMDS N-Z  DEFAULT  FEATURE
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log# 1 --  ROW 168 TO 186 OF 186
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.VSAM.CSI,VOL=SER=SER004  -----------------------------------------
>----->ver
 VERIFY FILE(SYS00303)

 EXAMINE NAME(SER07.VSAM.CSI)   INDEXTEST   DATATEST ERRORLIMIT(4)
IDC01700I INDEXTEST BEGINS
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC01701I DATATEST BEGINS
IDC01709I DATATEST COMPLETE - NO ERRORS DETECTED
IDC01708I 14 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 2113 RECORDS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 88 BYTES
IDC01722I 95 PERCENT FREE SPACE

PDS006I End of data set
PDS111I 2,113 physical blocks were input
PDS112I 88 characters in the largest physical block
PDS113I 24 characters per average physical block
```

Figure 161. Sample VERIFY Subcommand (VSAM)



```
 - DSN=OUTPUT.DSN,VOL=SER=SER004  ----------------
>------>Verify  INPUT

PDS006I End of data set
PDS110I 12 logical records were input
PDS111I 1 physical blocks were input
PDS112I 960 characters in the largest physical block
PDS113I 960 characters per average physical block

PDS117I Cumulative size is 960 characters
```

Figure 162. Sample VERIFY Subcommand (sequential)

VPRINT Subcommand

Purpose The VPRINT (VTAM print) subcommand prints a hardcopy list of a member. The TSO VPSPRINT or DSPRINT command is used (as generated during StarWarp installation); any desired VPSPRINT/DSPRINT operands may be added after the member name.

Example VPRINT   mema:memb r525

Syntax

```
VPRINT   memgroup printer
         [CLASS(c)                               ]
         [COL(col1:col2, ...)                    ]
         [COPIES(number)                         ]
         [DDNAME(ddname)                         ]
         [DIRECT(KEEP/DELETE)                    ]
         [EJECT/NOEJECT                          ]
         [FCB(fcbname)                           ]
         [FOLD(width)/TRUNCATE(width)            ]
         [FORMS(formname)                        ]
         [HOLD/NOHOLD                            ]
         [LINES(linenum1:linenum2)               ]
         [NOEJMESS                               ]
         [NOHEADER                               ]
         [NUM(loc,length)/SNUM(loc,length)/NONUM ]
         [PAGELEN(num)                           ]
         [SINGLE/DOUBLE/CCHAR                    ]
         [TERMINAL                               ]
         [TMARGIN(num)/BMARGIN(num)              ]
         [TRACE                                  ]
         [WTR(wtrname)                           ]

        Note: the above operands are for the VPSPRINT command.
```

Aliases VP, VPR, VPRI, VPRIN, VPRINT

Defaults CLASS(A), COPIES(1), NOEJECT, NOHOLD, NUM, SINGLE

Required memgroup, printer

Operands

memgroup identifies the member(s) to be printed. Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

printer identifies the VTAM printer which is to print this data set.
operands optional, may include any desired TSO VPSPRINT or DSPRINT operands.

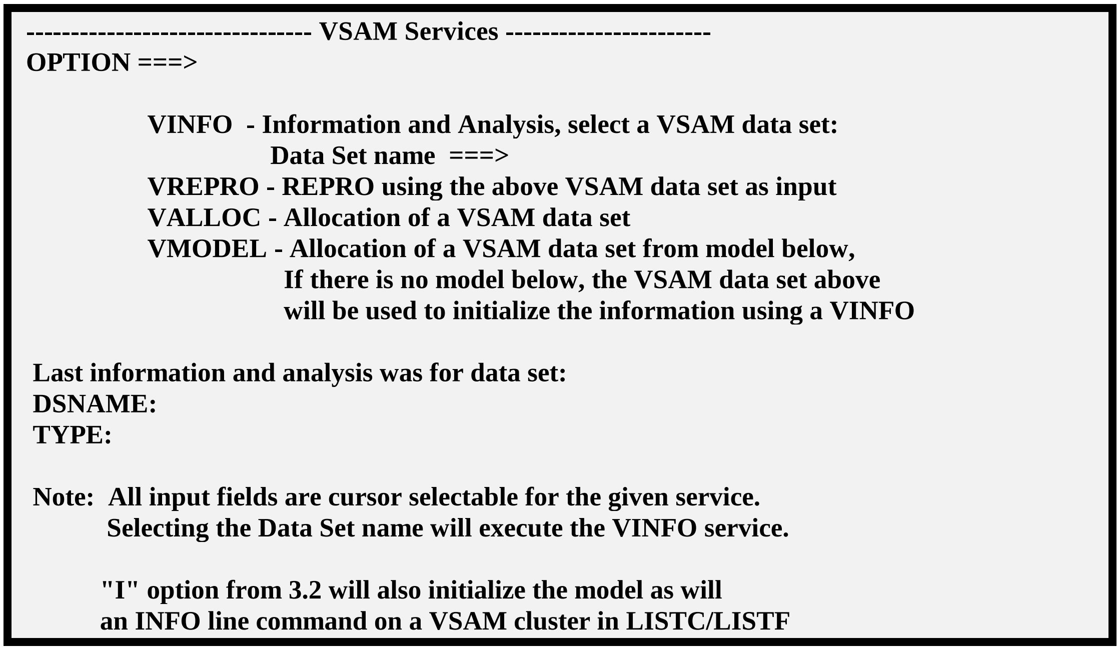
Remarks The VPRINT (VTAM print) subcommand prints a hardcopy list of a member.

Either the TSO VPSPRINT or DSPRINT command is used; this choice was made during StarWarp installation. To determine which program is used, enter a **CONTROL DEFAULT** subcommand and look for an output line beginning "VPRINT calls".

VSAM Subcommand

Purpose The VSAM subcommand invokes StarWarp VSAM Services.

Syntax   VSAM



```
------------------------------- VSAM Services -----------------------
OPTION ===>

            VINFO  - Information and Analysis, select a VSAM data set:
                        Data Set name  ===>
            VREPRO - REPRO using the above VSAM data set as input
            VALLOC - Allocation of a VSAM data set
            VMODEL - Allocation of a VSAM data set from model below,
                         If there is no model below, the VSAM data set above
                         will be used to initialize the information using a VINFO

 Last information and analysis was for data set:
 DSNAME:
 TYPE:

 Note:  All input fields are cursor selectable for the given service.
        Selecting the Data Set name will execute the VINFO service.

        "I" option from 3.2 will also initialize the model as will
        an INFO line command on a VSAM cluster in LISTC/LISTF
```

VTOC Subcommand

This interface is optional, it should only be used if your installation has installed the public domain VTOC command. For current information on the DSAT command, refer to the CBT web site at http://www.cbttape.org.

Purpose The VTOC subcommand displays selected data sets on a disk or set of disks. Each disk has a volume table of contents (VTOC); this table is searched for data sets that meet the specifications.

Example VTOC tso001 cont(mvs43)

Syntax

```
VTOC   (volumelist)  [LEVEL(dsnamestart)              ]
                     [EXLEVEL(dsnamestart)            ]
                     [CONTAINING(dsnamestring)        ]
                     [ENDING(dsnameend)               ]
                     [BREAK(breakchars)               ]
            [CAT                                      ]
            [TRK/CYL/KB/MB                            ]
            [NOSORT/SORT(sortfields)                  ]
            [NOPRINT/PRINT(printop (printitemlist))   ]
            [LIMIT(keyword oper value)                ]
            [AND1/OR1(keyword oper value)             ]
            [AND2/OR2(keyword oper value)             ]
            [AND3/OR3(keyword oper value)             ]
            [CHARS(charsperline)                      ]
            [LINES(linesperpage)                      ]
            [NOHEADING/HEADING(text)                  ]
            [DSNLEN(length)                           ]
```

Aliases VT, VTO, VTOC

Defaults List all data sets on the volume(s) selected; SORT, PRINT, TRK

Required volumelist

Operands

volumelist a volume name or a list of volume names in parentheses. If the first one to five characters of a volume name are entered, all volumes that are mounted which start with those characters will be listed. If ALL is specified, all non-virtual volumes which are online and ready will be processed. If ALV is specified, all virtual volumes which are online and ready will be processed, if they are mounted PRIVATE/RESERVED or PRIVATE/RESIDENT.

LEVEL(dsnstart) specifies the high level qualifiers to be searched. This will not be prefixed by your userid or prefix. Only data sets starting with these prefixes will be listed.

EXLEVEL(dsnstart) specifies the high level qualifiers to be omitted. This will not be prefixed by your userid or prefix. Data sets starting with this prefix will not be listed.

CONTAINING(dsnstr) specifies a character string contained in the data set name. At least one of the strings must be in the DSNAME for the data set to be listed. This string does not need to conform to DSNAME standards; it can begin with a period or a number.

ENDING(dsnameend) specifies the ending characters of the DSNAME. The final nonblank characters of the DSNAME must be one of these strings to allow the data set to be listed. This string must conform to DSNAME standards.

CAT A locate is done for each DSNAME on the volumes listed and status is indicated. Note: this option can take a considerable amount of processing time.

- **C** cataloged on this volume
- **N** not cataloged
- **W** cataloged on another volume
- **E** catalog processing error

TRK space is to be reported in track units.

CYL space is to be reported in cylinder units.

KB space is to be reported in kilobyte units.

MB space is to be reported in megabyte units.

NOSORT the data sets are not sorted. They are output as they are found.

SORT(sortfields) data sets are sorted into alphabetical order, based upon the sort fields specified. DSNAME, VOLUME, ALLOC, USED, UNUSED, PCT, EX, DSO, RFM, LRECL, BLKSZ, CDATE, EXPDTand REFDT are valid sort fields. A or D is required after each sort field to indicate ascending/descending sequence.

BREAK(breakchars) the listing will contain a new header, (on a new page if the VTOCOUT DD card option is used), whenever the specified number of characters differs from the preceding data set. This option functions only with the SORT option.

CHARS(charsperline) specifies the number of characters on each line of output. The default is 150 for print and the linesize of the terminal for TSO sessions. You can get more information by specifying a larger number of characters per line or you can limit the printing by setting a smaller number of characters per line of output.

LINES(linesperpage) specifies the number of lines before a new title line is produced. It defaults to 60 for print and to the screen size for TSO sessions.

NOHEADING do not produce a heading. The heading will only be output if the VTOCOUT DD statement is present.

HEADING(text) if a DD statement with a DDNAME of VTOCOUT is present, this text will be used to begin every page. ASA carriage control should be included. The default header consists of "VTOC command version 02" followed by the command that was entered.

DSNLEN(length) specifies the length of the DSNAME to print. The rest of the DSNAME is truncated. The CHARS parameter will also cause the DSNAME to be truncated, if the name and the preceding information exceeds the print line.

NOPRINT specifies that individual items are not to be listed. The command can be used to calculate totals.

PRINT(ptop (ptitems)) specifies the items to print. Ptop is the print formatting option as follows:

- **NEW** the ptitems is a complete list of what to print.
- **REP** the first ptitem will be replaced with the rest of the items on the list.
- **ADD** the rest of the ptitems will be added after the first item on the list.
- **DEL** the ptitems named will not be printed.

The ptitems names are the same keywords used in LIMIT, AND, and OR functions and are also the titles as printed. These keywords can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

The ADD, DEL, and REP print operations refer to the default print list. The default list is ALLOC, UNUSED, PCT, EX, DSO, RFM, BLKSZ, LRECL, REFDT, CDATE, VOLUME, DSNAME, EXPDT, SECQ, SECT, ROUND, PASS, ACTION, and type. As noted under CHARS above, only the items that will fit on the print line will be listed.

LIMIT(key oper value) specifies which data sets are to be listed. Only data sets that satisfy the relation are listed.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

keyword is the name of a data set field as follows. The keywords and their values are the same as in the VTOC output.

ALLOC data set allocation; number of kilobytes, tracks, cylinders, or megabytes allocated. Default is in tracks.

UNUSED amount of unused space in the data set. Same units as in ALLOC.

USED amount of space used in the data set. Same units as in ALLOC.

PCT percentage of space used in the data used.

EX number of extents in the data set.

DSO data set organization

PS = sequential
PO = partitioned
VS = VSAM
PE = PDSE data
DA = direct
IS = ISAM
U = unmoveable

RFM record format

F = fixed
V = variable
U = undefined
B = blocked
T = track overflow
S = spanned or standard
A = ASA carriage control
M = machine carriage control

BLKSZ blocksize for physical blocks of data.

LRECL logical record length in bytes.

CDATE creation date in the form YYDDD, sometimes called Julian.

EXPDT expiration date in the same form. This field is rarely used.

REFDT last use date in the same form. This date is when the data set was last opened.

SECT type of allocation

A = absolute track
B = blocks
T = tracks
C = cylinders

PASS protection indicators
N = none
T = read and write protection
W = write protection

ROUND space rounded up to cylinders
R = round
N = no round

CCHH cylinder and head address, in 4 or 8 hexadecimal digits. If 4 digits are used, only the cylinder is used for comparison, otherwise, the cylinder and track are compared.

VOLUME volume serial number or disk name

UNIT unit or device type

DSNAME name of the data set

ACTION some error indications

TYPE reserved for exit usage.

RACF RACF indicator
N = not indicated
Y = indicated

UPD updated since last backup
N = not updated
Y = updated

oper is an operator. The list of operators is as follows:
EQ is equal to
NE is not equal to
LE is less than or equal to
LT is less than
GE is greater than or equal to
GT is greater than

value gives the value of the item for comparison, such as FB, PS, R, or a number.

AND1(key oper value) specifies which data sets are to be listed. Both LIMIT and this condition must be true to allow the listing.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

OR1(key oper value) specifies which data sets are to be listed. Either LIMIT and this condition must be true to allow the listing.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

AND2(key oper value) specifies which data sets are to be listed. Both the previous result and this condition must be true to allow the listing.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

OR2(key oper value) specifies which data sets are to be listed. Either the previous result or this condition must be true to allow the listing.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

AND3(key oper value) specifies which data sets are to be listed. Both the previous result and this condition must be true to allow the listing.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

OR3(key oper value) specifies which data sets are to be listed. Either the previous result or this condition must be true to allow the listing.

key can be ALLOC, UNUSED, USED, PCT, EX, DSO, RFM, BLKSZ, LRECL, CDATE, EXPDT, REFDT, SECT, PASS, ROUND, CCHH, VOLUME, UNIT, DSNAME, ACTION, TYPE, RACF or UPD.

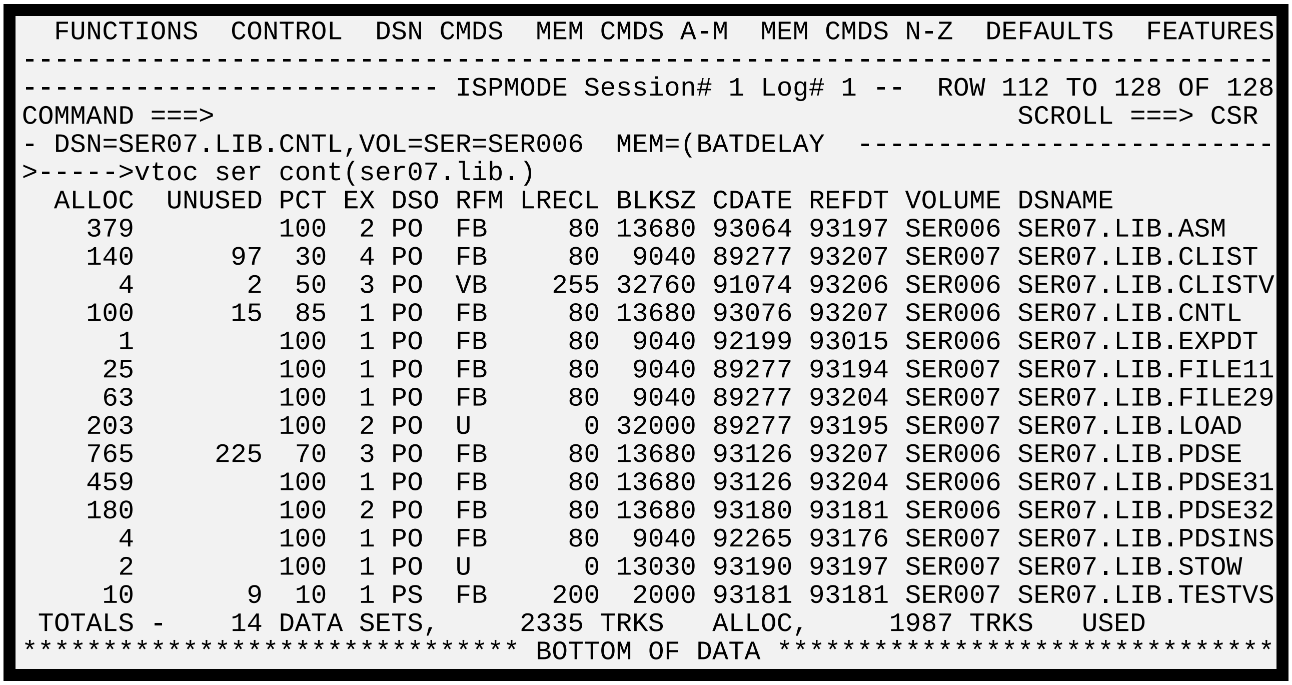
oper can be EQ, NE, LE, LT, GE or GT.

value can be a comparison value such as FB, PS, R or a number like 51.

Remarks Following are some examples of uses for the VTOC command.

1. List all data sets on VOL*** volumes:
   ```
   VTOC vol
   ```
2. List all data sets on all volumes:
   ```
   VTOC all
   ```
3. List all data sets that start with XXX on any MV**** volume:
   ```
   VTOC mv lev(xxx)
   ```
4. List all data sets that contain LIST and do not start with SYS1:
   ```
   VTOC vol cont(list) exlev(sys1)
   ```
5. List all recently created data sets:
   ```
   VTOC vol lim(cdate gt 93001)
   ```
6. List data sets with unused space:
   ```
   VTOC vol lim(pct lt 50)  or1(unused gt 30)
   ```
7. List data sets with multiple extents:
   ```
   VTOC vol lim(ex gt 1)
   ```
8. List all of a user's data sets on TSO001:
   ```
     VTOC tso001 level(ser002)
   ```

9. List all data sets over 100 tracks:
   `VTOC vol lim(alloc gt 100)`
10. List CLIST data sets:
    `VTOC vol end(clist)`
11. List all of the information about a user's data sets at a 3270:
    `VTOC vol char(150)`
12. List data sets under the fixed heads:
    `VTOC vol lim(cc eq 0001) or1(cc eq 0002)`
13. Check if data sets on volume VOL*** are cataloged correctly:
    `VTOC vol cat  print(new (alloc pct cat dsname))`
14. List used space instead of unused, in tracks:
    `VTOC vol print(rep (unused used))  trk`
15. List all data sets on 335*** volumes sorted by ALLOC in descending sequence, VOLUME and DSNAME in ascending sequence:
    `VTOC 335 sort(alloc,d,volume,a,dsname,a)`



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session# 1 Log# 1 --  ROW 112 TO 128 OF 128
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER006  MEM=(BATDELAY  -------------------------
>----->vtoc ser cont(ser07.lib.)
  ALLOC  UNUSED PCT EX DSO RFM LRECL BLKSZ CDATE REFDT VOLUME DSNAME
    379         100  2 PO  FB     80 13680 93064 93197 SER006 SER07.LIB.ASM
    140      97  30  4 PO  FB     80  9040 89277 93207 SER007 SER07.LIB.CLIST
      4       2  50  3 PO  VB    255 32760 91074 93206 SER006 SER07.LIB.CLISTV
    100      15  85  1 PO  FB     80 13680 93076 93207 SER006 SER07.LIB.CNTL
      1         100  1 PO  FB     80  9040 92199 93015 SER006 SER07.LIB.EXPDT
     25         100  1 PO  FB     80  9040 89277 93194 SER007 SER07.LIB.FILE11
     63         100  1 PO  FB     80  9040 89277 93204 SER007 SER07.LIB.FILE29
    203         100  2 PO  U       0 32000 89277 93195 SER007 SER07.LIB.LOAD
    765     225  70  3 PO  FB     80 13680 93126 93207 SER006 SER07.LIB.PDSE
    459         100  1 PO  FB     80 13680 93126 93204 SER006 SER07.LIB.PDSE31
    180         100  2 PO  FB     80 13680 93180 93181 SER006 SER07.LIB.PDSE32
      4         100  1 PO  FB     80  9040 92265 93176 SER007 SER07.LIB.PDSINS
      2         100  1 PO  U       0 13030 93190 93197 SER007 SER07.LIB.STOW
     10       9  10  1 PS  FB    200  2000 93181 93181 SER007 SER07.LIB.TESTVS
 TOTALS -    14 DATA SETS,     2335 TRKS   ALLOC,     1987 TRKS   USED
****************************** BOTTOM OF DATA ******************************
```

Figure 168. Sample VTOC Subcommand

VUSE Subcommand

Purpose The VUSE (volume usage) subcommand checks disk volume statistics and usage.

Example VUSE   sysres all

Syntax

```
VUSE  volume [SHORT/LONG/ALL]
```

Aliases VU, VUS, VUSE

Defaults SHORT

Required volume

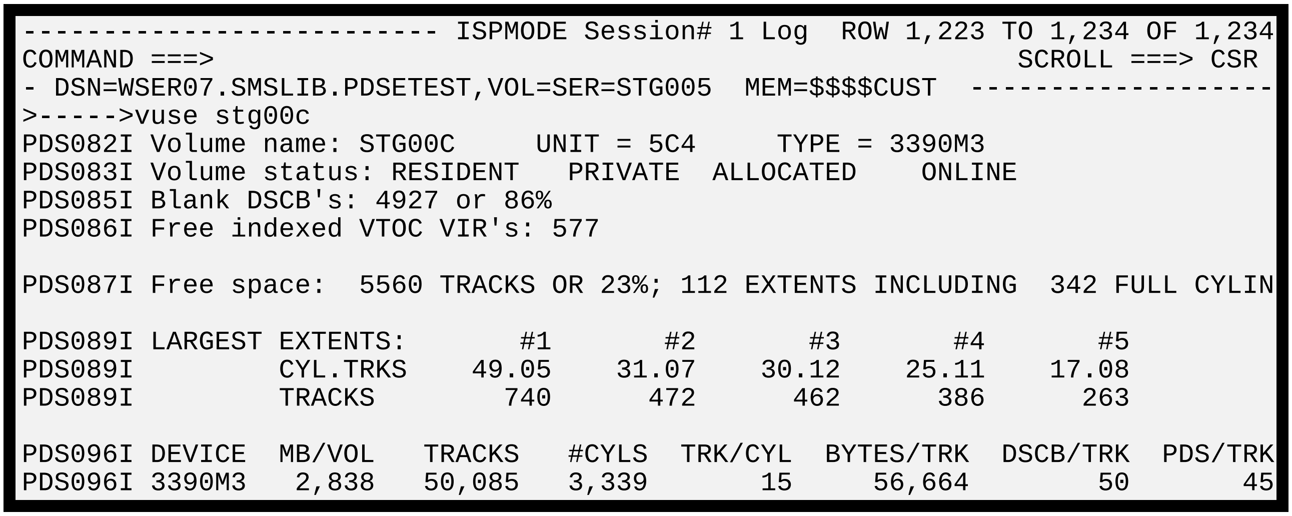
Operands

SHORT provide basic information for a volume including free space statistics and the current volume usage.

ALL include **SHORT** information and format any SMS VOLUME RECORD information and the FORMAT 4 DSCB. Same as LONG.

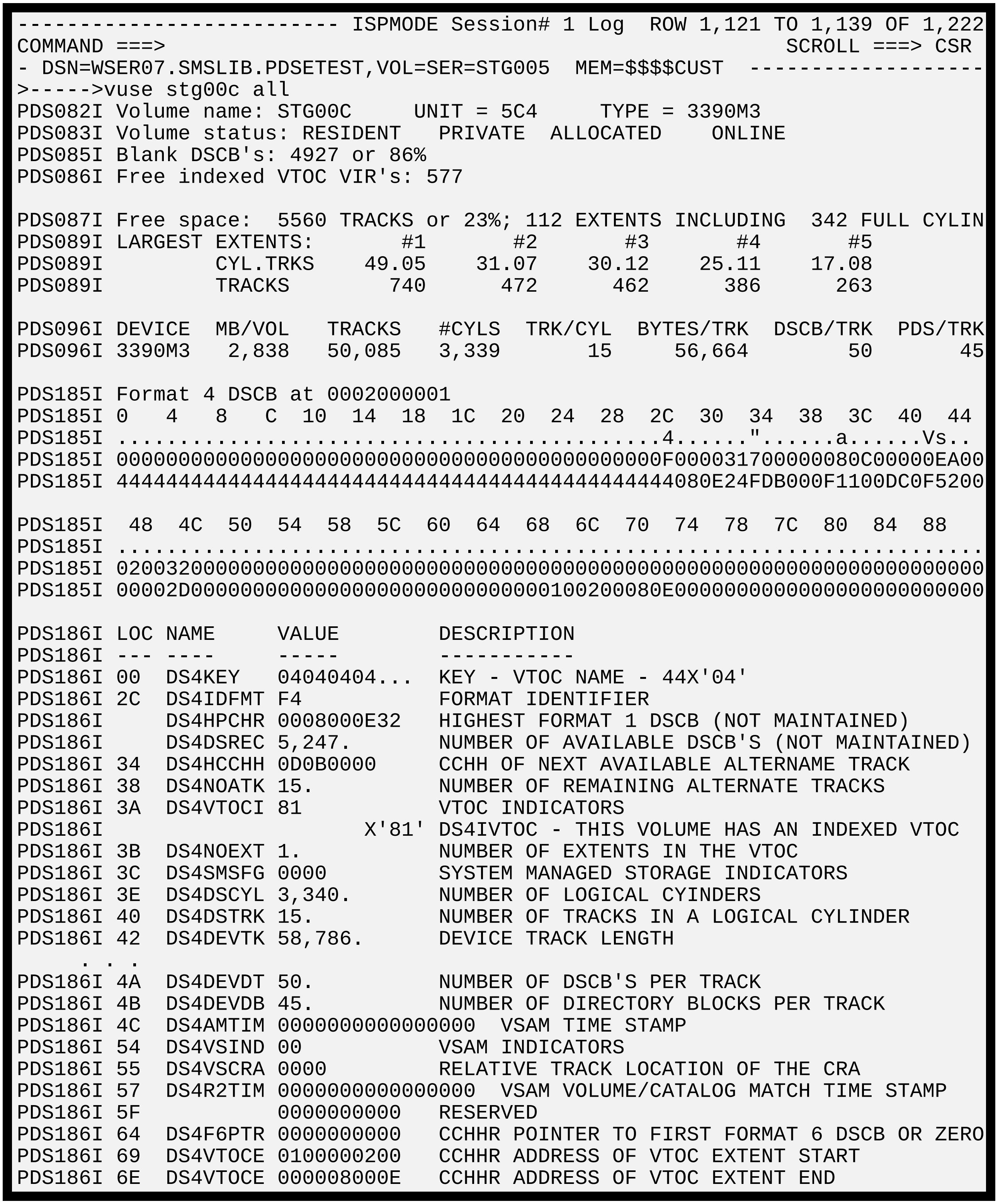
LONG include **SHORT** information and format any SMS VOLUME RECORD information and the FORMAT 4 DSCB. Same as ALL.

Remarks The VUSE subcommand is useful for determining the amount of free space on a volume as it displays the largest five free extents in track and cylinder units. It is also useful for diagnosing volume errors as the FORMAT 4 DSCB can be interpreted by this subcommand.



```
------------------------- ISPMODE Session# 1 Log  ROW 1,223 TO 1,234 OF 1,234
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.SMSLIB.PDSETEST,VOL=SER=STG005  MEM=$$$$CUST  -------------------
>----->vuse stg00c
PDS082I Volume name: STG00C     UNIT = 5C4     TYPE = 3390M3
PDS083I Volume status: RESIDENT   PRIVATE  ALLOCATED    ONLINE
PDS085I Blank DSCB's: 4927 or 86%
PDS086I Free indexed VTOC VIR's: 577

PDS087I Free space:  5560 TRACKS OR 23%; 112 EXTENTS INCLUDING  342 FULL CYLIN

PDS089I LARGEST EXTENTS:       #1       #2       #3       #4       #5
PDS089I         CYL.TRKS    49.05    31.07    30.12    25.11    17.08
PDS089I         TRACKS        740      472      462      386      263

PDS096I DEVICE  MB/VOL   TRACKS   #CYLS  TRK/CYL  BYTES/TRK  DSCB/TRK  PDS/TRK
PDS096I 3390M3   2,838   50,085   3,339       15     56,664        50       45
```

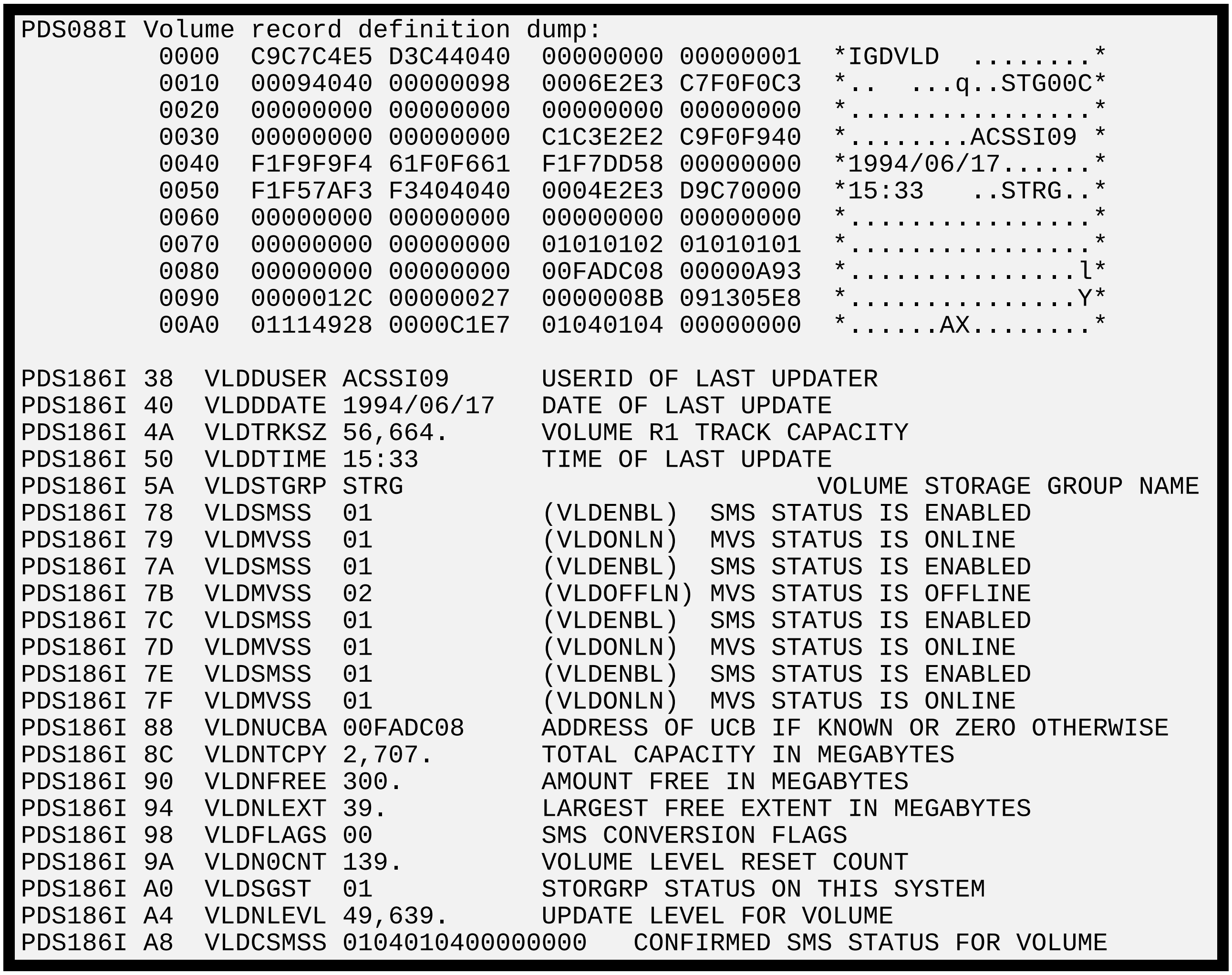
Figure 169. Sample VUSE Subcommand



```
-------------------------- ISPMODE Session# 1 Log  ROW 1,121 TO 1,139 OF 1,222
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=WSER07.SMSLIB.PDSETEST,VOL=SER=STG005  MEM=$$$$CUST  -------------------
>----->vuse stg00c all
PDS082I Volume name: STG00C     UNIT = 5C4     TYPE = 3390M3
PDS083I Volume status: RESIDENT   PRIVATE  ALLOCATED    ONLINE
PDS085I Blank DSCB's: 4927 or 86%
PDS086I Free indexed VTOC VIR's: 577

PDS087I Free space:  5560 TRACKS or 23%; 112 EXTENTS INCLUDING  342 FULL CYLIN
PDS089I LARGEST EXTENTS:       #1       #2       #3       #4       #5
PDS089I         CYL.TRKS    49.05    31.07    30.12    25.11    17.08
PDS089I         TRACKS        740      472      462      386      263

PDS096I DEVICE  MB/VOL   TRACKS   #CYLS  TRK/CYL  BYTES/TRK  DSCB/TRK  PDS/TRK
PDS096I 3390M3   2,838   50,085   3,339       15     56,664        50       45

PDS185I Format 4 DSCB at 0002000001
PDS185I 0   4   8   C  10  14  18  1C  20  24  28  2C  30  34  38  3C  40  44
PDS185I ............................................4......"......a.......Vs..
PDS185I 00000000000000000000000000000000000000000000F000031700000080C00000EA00
PDS185I 444444444444444444444444444444444444444444444080E24FDB000F1100DC0F5200

PDS185I  48  4C  50  54  58  5C  60  64  68  6C  70  74  78  7C  80  84  88
PDS185I ......................................................................
PDS185I 0200320000000000000000000000000000000000000000000000000000000000000000
PDS185I 00002D00000000000000000000000000000100200080E0000000000000000000000000

PDS186I LOC NAME     VALUE        DESCRIPTION
PDS186I --- ----     -----        -----------
PDS186I 00  DS4KEY   04040404...  KEY - VTOC NAME - 44X'04'
PDS186I 2C  DS4IDFMT F4           FORMAT IDENTIFIER
PDS186I     DS4HPCHR 0008000E32   HIGHEST FORMAT 1 DSCB (NOT MAINTAINED)
PDS186I     DS4DSREC 5,247.       NUMBER OF AVAILABLE DSCB'S (NOT MAINTAINED)
PDS186I 34  DS4HCCHH 0D0B0000     CCHH OF NEXT AVAILABLE ALTERNAME TRACK
PDS186I 38  DS4NOATK 15.          NUMBER OF REMAINING ALTERNATE TRACKS
PDS186I 3A  DS4VTOCI 81           VTOC INDICATORS
PDS186I                     X'81' DS4IVTOC - THIS VOLUME HAS AN INDEXED VTOC
PDS186I 3B  DS4NOEXT 1.           NUMBER OF EXTENTS IN THE VTOC
PDS186I 3C  DS4SMSFG 0000         SYSTEM MANAGED STORAGE INDICATORS
PDS186I 3E  DS4DSCYL 3,340.       NUMBER OF LOGICAL CYINDERS
PDS186I 40  DS4DSTRK 15.          NUMBER OF TRACKS IN A LOGICAL CYLINDER
PDS186I 42  DS4DEVTK 58,786.      DEVICE TRACK LENGTH
     . . .
PDS186I 4A  DS4DEVDT 50.          NUMBER OF DSCB'S PER TRACK
PDS186I 4B  DS4DEVDB 45.          NUMBER OF DIRECTORY BLOCKS PER TRACK
PDS186I 4C  DS4AMTIM 0000000000000000  VSAM TIME STAMP
PDS186I 54  DS4VSIND 00           VSAM INDICATORS
PDS186I 55  DS4VSCRA 0000         RELATIVE TRACK LOCATION OF THE CRA
PDS186I 57  DS4R2TIM 0000000000000000  VSAM VOLUME/CATALOG MATCH TIME STAMP
PDS186I 5F           0000000000   RESERVED
PDS186I 64  DS4F6PTR 0000000000   CCHHR POINTER TO FIRST FORMAT 6 DSCB OR ZERO
PDS186I 69  DS4VTOCE 0100000200   CCHHR ADDRESS OF VTOC EXTENT START
PDS186I 6E  DS4VTOCE 000008000E   CCHHR ADDRESS OF VTOC EXTENT END
```

Figure 170. Sample VUSE with ALL (Non-SMS)

If a volume is SMS-controlled, the ALL or LONG operand also formats the following SMS VOLUME RECORD information.



```
PDS088I Volume record definition dump:
         0000  C9C7C4E5 D3C44040  00000000 00000001  *IGDVLD  ........*
         0010  00094040 00000098  0006E2E3 C7F0F0C3  *..  ...q..STG00C*
         0020  00000000 00000000  00000000 00000000  *................*
         0030  00000000 00000000  C1C3E2E2 C9F0F940  *........ACSSI09 *
         0040  F1F9F9F4 61F0F661  F1F7DD58 00000000  *1994/06/17......*
         0050  F1F57AF3 F3404040  0004E2E3 D9C70000  *15:33   ..STRG..*
         0060  00000000 00000000  00000000 00000000  *................*
         0070  00000000 00000000  01010102 01010101  *................*
         0080  00000000 00000000  00FADC08 00000A93  *...............l*
         0090  0000012C 00000027  0000008B 091305E8  *...............Y*
         00A0  01114928 0000C1E7  01040104 00000000  *......AX........*

PDS186I 38  VLDDUSER ACSSI09      USERID OF LAST UPDATER
PDS186I 40  VLDDDATE 1994/06/17   DATE OF LAST UPDATE
PDS186I 4A  VLDTRKSZ 56,664.      VOLUME R1 TRACK CAPACITY
PDS186I 50  VLDDTIME 15:33        TIME OF LAST UPDATE
PDS186I 5A  VLDSTGRP STRG                               VOLUME STORAGE GROUP NAME
PDS186I 78  VLDSMSS  01           (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 79  VLDMVSS  01           (VLDONLN)  MVS STATUS IS ONLINE
PDS186I 7A  VLDSMSS  01           (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 7B  VLDMVSS  02           (VLDOFFLN) MVS STATUS IS OFFLINE
PDS186I 7C  VLDSMSS  01           (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 7D  VLDMVSS  01           (VLDONLN)  MVS STATUS IS ONLINE
PDS186I 7E  VLDSMSS  01           (VLDENBL)  SMS STATUS IS ENABLED
PDS186I 7F  VLDMVSS  01           (VLDONLN)  MVS STATUS IS ONLINE
PDS186I 88  VLDNUCBA 00FADC08     ADDRESS OF UCB IF KNOWN OR ZERO OTHERWISE
PDS186I 8C  VLDNTCPY 2,707.       TOTAL CAPACITY IN MEGABYTES
PDS186I 90  VLDNFREE 300.         AMOUNT FREE IN MEGABYTES
PDS186I 94  VLDNLEXT 39.          LARGEST FREE EXTENT IN MEGABYTES
PDS186I 98  VLDFLAGS 00           SMS CONVERSION FLAGS
PDS186I 9A  VLDN0CNT 139.         VOLUME LEVEL RESET COUNT
PDS186I A0  VLDSGST  01           STORGRP STATUS ON THIS SYSTEM
PDS186I A4  VLDNLEVL 49,639.      UPDATE LEVEL FOR VOLUME
PDS186I A8  VLDCSMSS 0104010400000000   CONFIRMED SMS STATUS FOR VOLUME
```

Figure 171. Sample VUSE with ALL (SMS Volume Information)

WHOHAS Subcommand

Purpose The WHOHAS subcommand displays users allocated to a data set.

Example WHOHAS 'sys1.uads'

Syntax

```
WHOHAS   dataset / *
```

Aliases WH, WHO, WHOH, WHOHA, WHOHAS

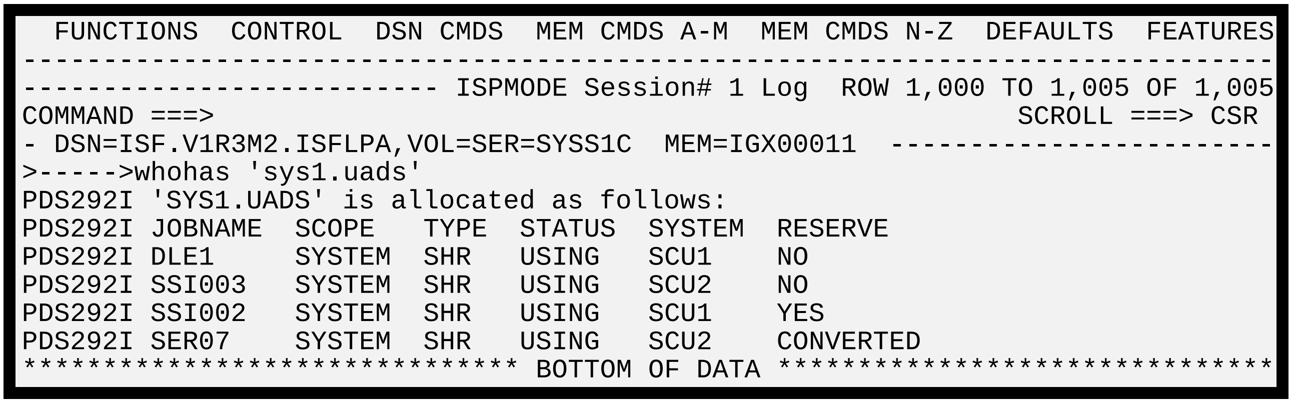
Defaults none

Required dataset

Operands

dataset identifies the data set to be checked.
***** specifies that the currently allocated data set is to be checked.

Remarks WHO (short for WHOHAS) can be used as a line command in LISTA/DDNAME, LISTC/LISTF and WORKPAD.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
-------------------------- ISPMODE Session# 1 Log  ROW 1,000 TO 1,005 OF 1,005
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=ISF.V1R3M2.ISFLPA,VOL=SER=SYSS1C  MEM=IGX00011  -----------------------
>----->whohas 'sys1.uads'
PDS292I 'SYS1.UADS' is allocated as follows:
PDS292I JOBNAME  SCOPE   TYPE  STATUS  SYSTEM  RESERVE
PDS292I DLE1     SYSTEM  SHR   USING   SCU1    NO
PDS292I SSI003   SYSTEM  SHR   USING   SCU2    NO
PDS292I SSI002   SYSTEM  SHR   USING   SCU1    YES
PDS292I SER07    SYSTEM  SHR   USING   SCU2    CONVERTED
******************************* BOTTOM OF DATA ********************************
```

Figure 172. Sample WHOHAS Subcommand

WORKPAD Function

Purpose

The WORKPAD command is used to store STARWARP subcommands, data set names, TSO commands, CLISTS and REXX execs in an ISPF table. Each individual table element may be reexecuted when you wish.

WORKPAD tables are normally saved permanently. They are saved in the ISPF profile data set by default (ISPPROF) but the DDNAME can be changed in the SETALL option. To manage these tables, use the SAVE, ID, MERGE, NOSAVE and ERASE commands. To automatically save these tables at program end or when a new WORKPAD table is retrieved, check the setting of "Automatic save" in SETALL for WORKPAD.

Saved WORKPAD tables are given members names of the form **PDSWP0na** if the name is one or two numeric digits; otherwise, a name of the form **@@name** is used where name is a one to six character alphameric table identification name.

Example

WORKPAD

Syntax

```
WORKPAD   [name]  [ALTERNAT]  [SETUP]
```

Aliases

W, WO, WOR, WORK, WORKP, WORKPA, WORKPAD

Defaults

0

Required

none

Operands

name	one to six alphameric characters, the WORKPAD table name to retrieve.
ALTERNAT	display the alternate WORKPAD panel (for new WORKPADs only).
SETUP	execute each line of the WORKPAD in EXPRESS mode.

Remarks

The WORKPAD ISPF table is displayed in response to a WORKPAD command. When you are in a WORKPAD display, you have many options: you may delete a part of the table, find data in the table, insert elements into the table, print a part of the table, store a part of the table in a data set, ... For assistance with the different options available, you may use the HELP command, CUA functions or you may enter an O command as either a primary command or as a line command.

WORKPAD is designed to be a versatile service to keep sets of commands and data set names for easy reference. These sets can be shared between groups and the installation. Sample table 99 is supplied by SERENA and may be modified by your installation as an index to other samples. It is recommended that you reserve table 0 for temporary use and table 1 for your own index. Table 2 can be used for setup.

The SETUP operand on the WORKPAD command allows you to execute a series of STARWARP commands without intervention. Any data set entries in this mode will result in executing a GO line command and each data set entry can be followed by a MEMLIST command to build a MEMLIST for each data set.

Note that when a WORKPAD table entry is modified, it is not normally executed automatically; however, you can change this default for the current session by entering **MODE EXEC**
You can also change this default across sessions in **SETALL**.

Each WORKPAD element or line has the following four fields:
OPT option, specifies the program action desired (the line command field).
MSG/RC message or return code, displays program feedback messages.
TYPE command type, specifies the type of entry.

- ***** is a comment entry.
- **-** is a TSO command, a CLIST or a REXX exec.
- **D** is a data set name.
- **P** is a STARWARP subcommand.
- **T** is a TSO command whose output is to be TRAPPED in the log.

PDS/TSO command field, this is where the data set name, command or subcommand is entered and displayed. Note that comments may be added after any entry; just enter /* and follow it with any data.

The alternate WORKPAD table contains additional entry fields for data sets. You may enter the volume name and a member group specification with up to 17 characters if desired. The BB block line command and the BR, ED and ML line commands use the MEMBERS field from the alternate WORKPAD table.

The following primary commands are supported directly for the WORKPAD function; for documentation on ISPMODE commands available anywhere in STARWARP, see **Common Commands** on page 91.

ADD adds 5 blank table lines to the end of the WORKPAD table.
ALT[ERNAT] displays an alternate view of the WORKPAD table
APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`
DUA[L] displays a double line view of the WORKPAD table.
EDITT[BL] (or **ET[BL]**) enters an edit session on WORKPAD table data.
ER[ASE] deletes the WORKPAD table in memory and on disk.
EXPR[ESS] executes all entered line commands without pauses between individual commands.
F finds a string and positions the display start location.
Syntax:
```
F anystring [ASIS]
   [FIRST/LAST/PREV]    [PREFIX/SUFFIX/WORD]
   [ANY/COMMAND]
```

FI[ND] global command, changes to each data set in the table and issues a FIND subcommand. Since the syntax entered on a FIND subcommand is actually applied to each data set individually, you should not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.
Syntax:
```
FIND memgroup 'anystring'
      [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
      [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
```
GL[OBAL] global command, changes to each data set in the table and issues any STARWARP subcommand.
Syntax: `GLOBAL anysubcommand [anyoperands]`
ID changes the table ID name.
Syntax: `ID name` (where name is a one to six character alphameric name).

INS[ERT] adds blank WORKPAD table lines.

Syntax: `INSERT [nn] [TOP/BOTTOM/HERE]` (where nn defaults to 5)

LE[FT] rotates through alternate views of the WORKPAD table. PF keys 10 and 22 are normally set to LEFT.

LO[AD] loads data set names from operating system control blocks.

Syntax: `LOAD APFLST/LNKLST/LPALST [RESET]`

`Note: LOAD LNKLST now supports both static and dynamic linklists.`

MER[GE] merges data from a different table into the current table.

```
Syntax: MERGE name [GROUP/SAMPLE] [RESET]
                [TOP/BOTTOM/HERE]
```

(where name is one to six alphameric characters)

MODE specifies if modified WORKPAD lines are to be automatically executed.

Syntax: `MODE [EXEC/NOEXEC]`

MODEL global command, changes to each data set in the table and issues a MODEL command.

NOR[MAL] displays the default view of the WORKPAD table.

NOS[AVE] specifies that the current WORKPAD should not be saved on disk regardless of the setting of "Automatic save" in SETALL for WORKPAD.

O[PTIONS] provides primary command selection for the WORKPAD function and operand syntax assistance.

OUT[PUT] outputs the WORKPAD table to print or a data set.

Syntax: `OUTPUT [=c / F(ddname)]`

REM[OVE] trims the WORKPAD table based on a string match.

```
Syntax: REMOVE anystring [PREFIX/SUFFIX/WORD] [NOT]
                [ANY/COMMAND]
```

REPL[ACE] global command, changes to each data set in the table and issues a REPLACE subcommand. Since the syntax entered on a REPLACE subcommand is actually applied to each data set individually, you should not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.

```
Syntax: REPLACE memgroup 'fromstring' 'tostring'
                [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
                [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
                [WRITE/NOWRITE]
```

RESET often used with data set tagging, clears the DATA/MSG field in all table entries.

RF[IND] finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the WORKPAD table. PF keys 11 and 23 are normally set to RIGHT.

SAM[PLE] merges table number 99 from the installation table library (usually ISPTLIB) into the current WORKPAD table. Table 99 is an index to other sample WORKPAD tables. The additional sample tables can be included by using the MERGE command.

SAVE creates a permanent table for use in a later STARWARP session.

Syntax: `SAVE [name] [REPLACE/NOREPL]`

(where name is one to six alphameric characters)

SEEK global command, changes to each data set in the table and looks for a member.

Syntax: `SEEK member`

SO[RT] sorts the WORKPAD table into order based on the WORKPAD entries.

Syntax: `SORT`

TAG applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.

Syntax: `TAG linecmd`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation.
WHO[HAS] global command, issues a WHOHAS subcommand for each data set in the table.
X clears the WORKPAD table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`
XA[LL] clears the WORKPAD table; this is equivalent to **X ALL**

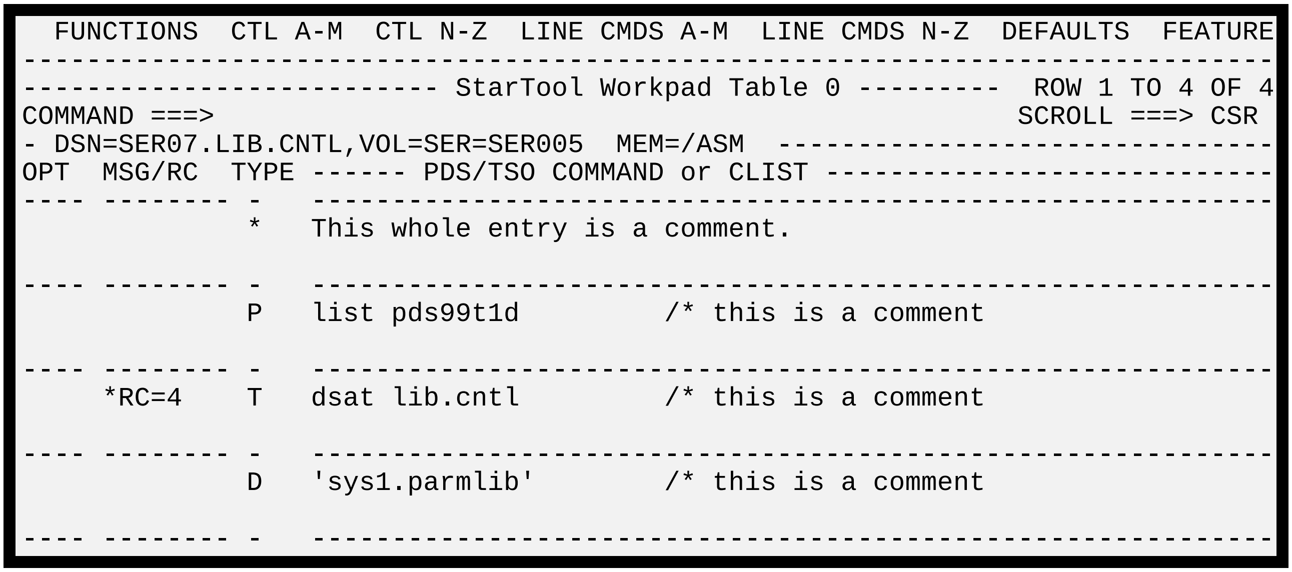
The following line commands are supported in the WORKPAD function. Note that the DFHSM commands are supported from the UT panel and that BR, ED and ML refer to a MEMBERS column which is displayed on an alternate panel which can be reached with PF10 or PF11.

+ provide an extension panel for command entry.
= repeat the previous line command.
6 change to the data set and MEMLIST all.
A after command, copy or move data after the marker.
ALT change to the data set and execute an alternate subcommand.
B before command, copy or move data before the marker.
BR change to the data set and MEMLIST for members in the MEMBERS column.
C copy this line.
CH change to the data set.
ED change to the data set and MEMLIST for members in the MEMBERS column.
EXEC execute the command, subcommand, CLIST, REXX exec or change to the data set.
FIND change to the data set and invoke FIND.
HBAC HBBACK DFHSM command to back up the data set
HBDE HBDEL DFHSM command for the data set.
HDEL HDEL DFHSM command for the data set.
HIST change to the data set and invoke HISTORY.
HMIG HMIG DFHSM command for the data set.
HML2 HMIG DFHSM command for the data set to LEVEL 2.
HREC HRECALL DFHSM command for the data set.
GO change to the data set using GO processing (a number is optional).
IF change to the data set and invoke IF.
IN insert a blank line.
K kill and clear all following line commands.
LC add the data set name to the current LISTC/LISTF table.
LOG copy the line into the log.
M move this line.
MAP change to the data set and invoke MAP.
ME provide line command selection and entry assistance.
ML change to the data set and MEMLIST for members in the MEMBERS column.
MOD change to the data set and MODEL.
O provide line command selection and operand syntax assistance.
PBRO change to the data set and invoke PBROWSE.
R reproduce this line.
REPL change to the data set and invoke REPLACE.
S select line command (normally EXEC, it is set in SETSEL).
SEEK change to the data set and check for a member.
SETA save a STARWARP subcommand for repeated use by the ALT line command.
SUBL change to the data set and invoke SUBLIST.
TAG mark this table entry with *TAG* in the DATA/MSG field.
UP update or view; then execute this entry.
US change to the data set and display USAGE.
UT select the extended user line command panel.
VERI change to the data set and invoke VERIFY.

VIEW view and update but do not execute this entry.
WHO check for users of this data set with WHOHAS.
X drop the table line.

Block line commands are doubled letters which delimit a range of table entries for which the same action is to be performed. The following block line commands are supported in the WORKPAD function:

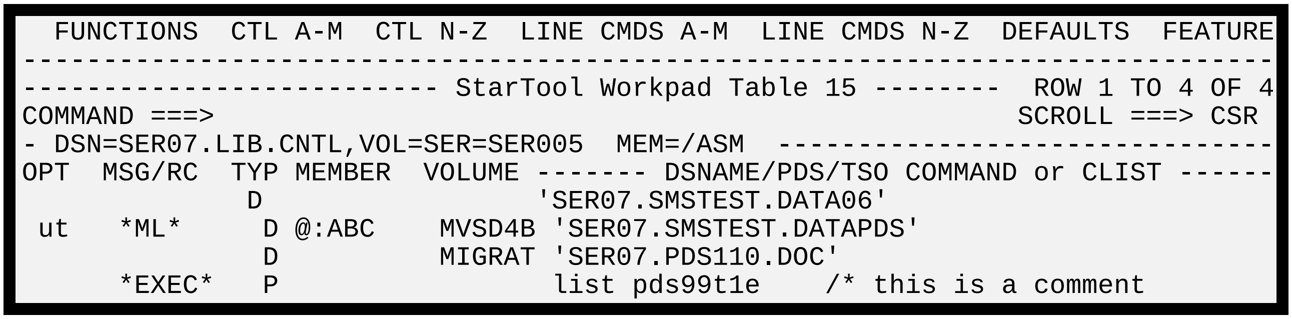
== = command, repeat the previous line command for the range of lines.
AA ALT command, change to each data set and execute an alternate subcommand.
BB BR command, change to each data set and MEMLIST for the MEMBERS specified.
CC C command, copy this range of table entries.
EE EXEC command, execute each entry.
GG GO command, change to each data set using GO processing.
LL LOG command, copy the range of lines into the log.
MM M command, move this range of table entries.
OO O command, provide line command assistance for each line.
RR R command, reproduce this range of table entries.
SS S command, select each line in the range of table lines.
SSEE SEEK command, change to each data set and check for a member.
UU UP command, update or view and execute each entry.
VV VIEW command, view and change each entry but do not execute.
XX X command, drop the range of table lines.



```
  FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------------------------------------------------------------
------------------------- StarTool Workpad Table 0 ---------  ROW 1 TO 4 OF 4
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER005  MEM=/ASM  -------------------------------
OPT  MSG/RC  TYPE ------ PDS/TSO COMMAND or CLIST ----------------------------
---- -------- -   ------------------------------------------------------------
              *   This whole entry is a comment.

---- -------- -   ------------------------------------------------------------
              P   list pds99t1d         /* this is a comment

---- -------- -   ------------------------------------------------------------
     *RC=4    T   dsat lib.cntl         /* this is a comment

---- -------- -   ------------------------------------------------------------
              D   'sys1.parmlib'        /* this is a comment

---- -------- -   ------------------------------------------------------------
```

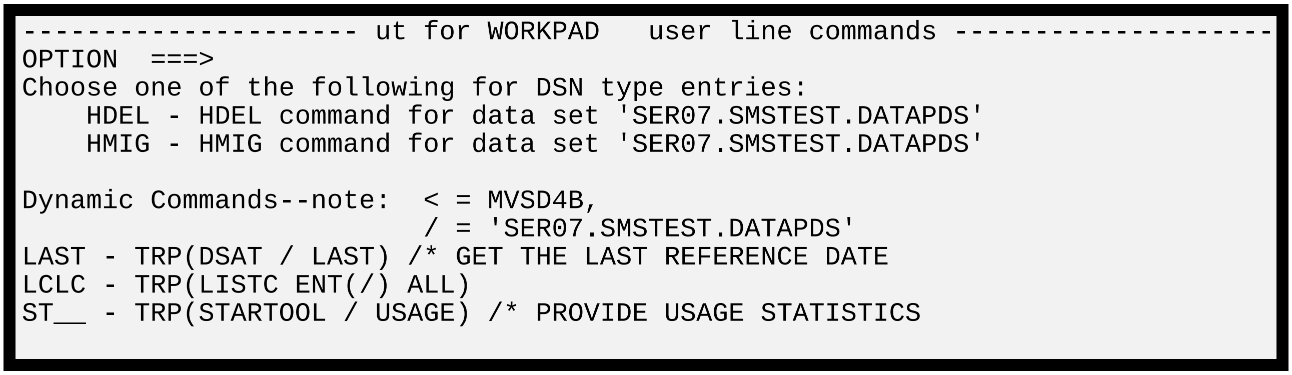
Figure 173. Sample WORKPAD Table

WORKPAD Function



```
  FUNCTIONS  CTL A-M  CTL N-Z  LINE CMDS A-M  LINE CMDS N-Z  DEFAULTS  FEATURE
------------------------------------------------------------------------------
-------------------------- StarTool Workpad Table 15 --------  ROW 1 TO 4 OF 4
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=SER07.LIB.CNTL,VOL=SER=SER005  MEM=/ASM  ------------------------------
OPT  MSG/RC  TYP MEMBER  VOLUME ------- DSNAME/PDS/TSO COMMAND or CLIST ------
              D                 'SER07.SMSTEST.DATA06'
 ut   *ML*     D @:ABC    MVSD4B 'SER07.SMSTEST.DATAPDS'
               D          MIGRAT 'SER07.PDS110.DOC'
      *EXEC*   P                 list pds99t1e    /* this is a comment
```

Figure 174. Sample WORKPAD Table (alternate)



```
--------------------- ut for WORKPAD   user line commands --------------------
OPTION  ===>
Choose one of the following for DSN type entries:
    HDEL - HDEL command for data set 'SER07.SMSTEST.DATAPDS'
    HMIG - HMIG command for data set 'SER07.SMSTEST.DATAPDS'

Dynamic Commands--note:  < = MVSD4B,
                         / = 'SER07.SMSTEST.DATAPDS'
LAST - TRP(DSAT / LAST) /* GET THE LAST REFERENCE DATE
LCLC - TRP(LISTC ENT(/) ALL)
ST__ - TRP(STARTOOL / USAGE) /* PROVIDE USAGE STATISTICS
```

Figure 175. Sample WORKPAD user line command

XREF Subcommand

Purpose The XREF subcommand lists internal symbol cross references in a load module. It is similar in nature to the output produced by the XREF option of the linkage editor except that references are provided by actual name used and in order sorted by name or by location.

Example XREF   mema:memb

Syntax

```
XREF   memgroup [MODULE({* / Fullm / Partm*)} ]
                [ENTRY(Entname)               ]
                [SHORT                        ]
                [SORT/NOSORT                  ]
                [NOSTACK                      ]
```

Aliases XR, XRE, XREF

Defaults memgroup, SORT

Required none

Operands

memgroup identifies the load member(s) for which cross reference information is desired.

Default member names, member lists, member name ranges and member name patterns are allowed; for more information, see **Appendix A. Member Name Forms** on page 91.

MODULE(name) specifies a 1 to 8 byte partial external name which limits CSECT and ENTRY names for XREF reporting.

The MODULE operand has several valid forms:

MODULE(*) use the previous name entered on any MODULE keyword.
MODULE(Fullm) report only on CSECT or ENTRY FULLM.
MODULE(Partm*) report only on CSECT or ENTRY PARTM...

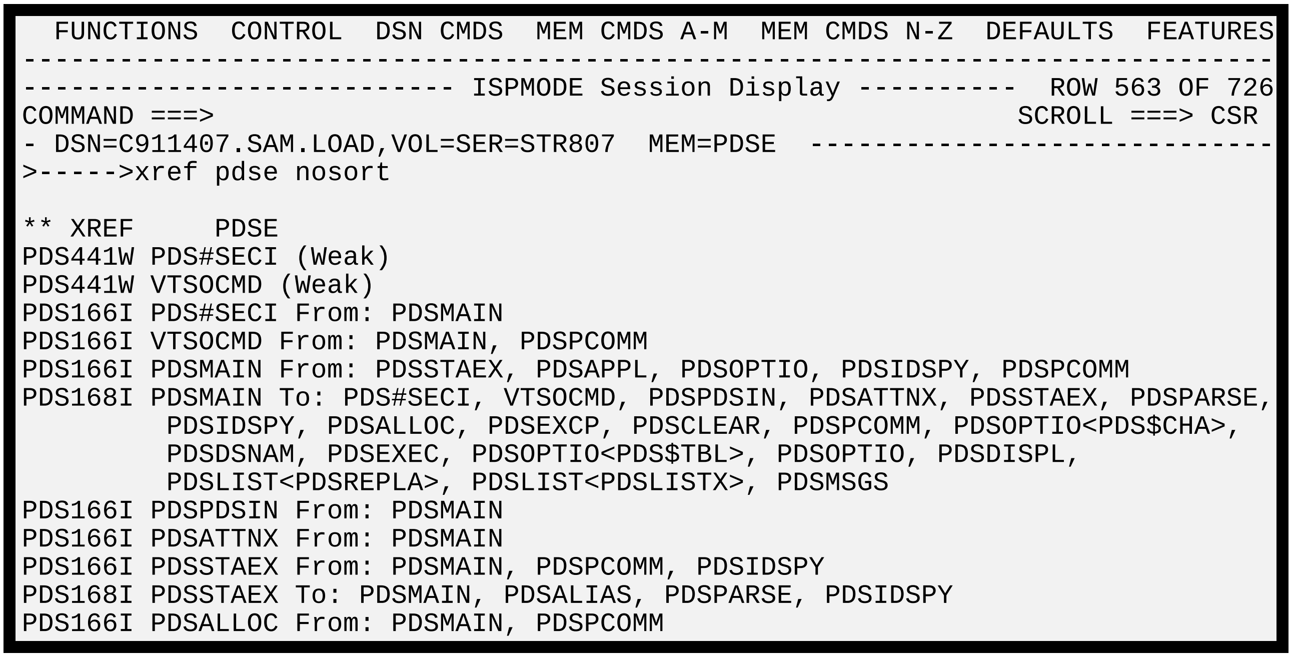
ENTRY(Entname) specifies a 1 to 8 byte partial entry name which may be used to limit XREF output similar to the MODULE keyword.
SHORT displays only the callers of a CSECT or ENTRY point name.
SORT specifies that CSECT names are to be output in order by CSECT name.
NOSORT specifies that CSECT names are to be output in order by address or CSECT location.
NOSTACK outputs a single data value on an output line for use with programs which post-process XREF outputs.

XREF Subcommand

Remarks The XREF subcommand lists internal symbol cross references in a load module.

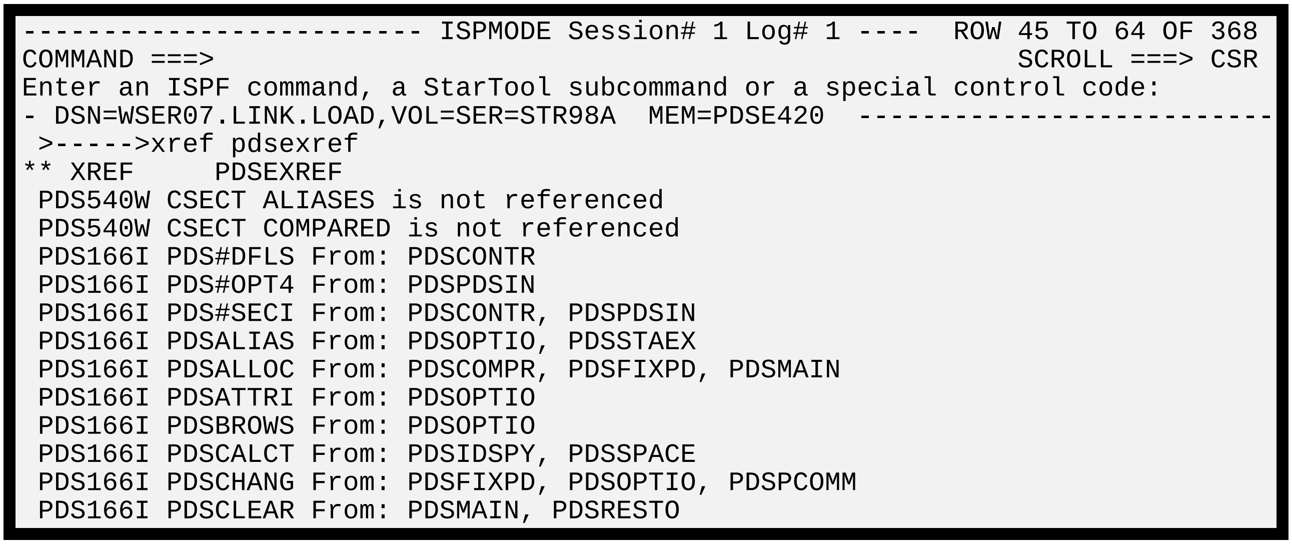
In the following example, the PDS441W message documents a missing weak external reference. Note that the missing CSECT (VTSOCMD) is referenced from PDSMAIN and PDSPCOMM.

Also, on the PDS168I message, the CSECT name and any ENTRY name referenced is noted with a special syntax. In this example, PDSMAIN has at least one reference to ENTRY PDS$CHA in CSECT PDSOPTIO; this information is represented by PDSOPTIO<PDS$CHA> in the second line of the first PDS168I message.



```
  FUNCTIONS  CONTROL  DSN CMDS  MEM CMDS A-M  MEM CMDS N-Z  DEFAULTS  FEATURES
------------------------------------------------------------------------------
------------------------- ISPMODE Session Display ----------  ROW 563 OF 726
COMMAND ===>                                                  SCROLL ===> CSR
- DSN=C911407.SAM.LOAD,VOL=SER=STR807  MEM=PDSE  -----------------------------
>----->xref pdse nosort

** XREF     PDSE
PDS441W PDS#SECI (Weak)
PDS441W VTSOCMD (Weak)
PDS166I PDS#SECI From: PDSMAIN
PDS166I VTSOCMD From: PDSMAIN, PDSPCOMM
PDS166I PDSMAIN From: PDSSTAEX, PDSAPPL, PDSOPTIO, PDSIDSPY, PDSPCOMM
PDS168I PDSMAIN To: PDS#SECI, VTSOCMD, PDSPDSIN, PDSATTNX, PDSSTAEX, PDSPARSE,
         PDSIDSPY, PDSALLOC, PDSEXCP, PDSCLEAR, PDSPCOMM, PDSOPTIO<PDS$CHA>,
         PDSDSNAM, PDSEXEC, PDSOPTIO<PDS$TBL>, PDSOPTIO, PDSDISPL,
         PDSLIST<PDSREPLA>, PDSLIST<PDSLISTX>, PDSMSGS
PDS166I PDSPDSIN From: PDSMAIN
PDS166I PDSATTNX From: PDSMAIN
PDS166I PDSSTAEX From: PDSMAIN, PDSPCOMM, PDSIDSPY
PDS168I PDSSTAEX To: PDSMAIN, PDSALIAS, PDSPARSE, PDSIDSPY
PDS166I PDSALLOC From: PDSMAIN, PDSPCOMM
```

Figure 176. Sample XREF Subcommand (with nosort)



```
------------------------ ISPMODE Session# 1 Log# 1 ----  ROW 45 TO 64 OF 368
COMMAND ===>                                                  SCROLL ===> CSR
Enter an ISPF command, a StarTool subcommand or a special control code:
- DSN=WSER07.LINK.LOAD,VOL=SER=STR98A  MEM=PDSE420  --------------------------
 >----->xref pdsexref
** XREF     PDSEXREF
 PDS540W CSECT ALIASES is not referenced
 PDS540W CSECT COMPARED is not referenced
 PDS166I PDS#DFLS From: PDSCONTR
 PDS166I PDS#OPT4 From: PDSPDSIN
 PDS166I PDS#SECI From: PDSCONTR, PDSPDSIN
 PDS166I PDSALIAS From: PDSOPTIO, PDSSTAEX
 PDS166I PDSALLOC From: PDSCOMPR, PDSFIXPD, PDSMAIN
 PDS166I PDSATTRI From: PDSOPTIO
 PDS166I PDSBROWS From: PDSOPTIO
 PDS166I PDSCALCT From: PDSIDSPY, PDSSPACE
 PDS166I PDSCHANG From: PDSFIXPD, PDSOPTIO, PDSPCOMM
 PDS166I PDSCLEAR From: PDSMAIN, PDSRESTO
```

Figure 177. Sample XREF Subcommand (sorted)

Commands

This section documents all commands in alphabetical order. These commands are supported in ISPMODE by various function commands as discussed in Subcommands and Functions on page 6. To look up any command, you may look in this chapter or check the index under "Command". The description of each function contains documentation on all commands specific to that function and all line commands and block line commands supported by that function.

In this chapter, defaults are shown with underscores and the shortest valid abbreviation for a command name is shown before the [brackets]. For example, a command shown as **COMM[ENT]** could be specified as COMM, COMME, COMMEN or COMMENT.

Common Commands

This first section of this chapter documents ISPMODE only commands which may be entered in any ISPMODE function. The next section of this chapter (see **All Commands** on page 91) documents all ISPMODE only commands which may be entered (including commands specific to a single function); these are referenced in Index entry "Command".

The following commands are supported by STARWARP anywhere in ISPMODE:

*****	merges current member group members into the MEMLIST table.
?	provides extended help on up to five warning or error messages from the last subcommand.
/*	allows any following information to be entered as comments.
ALIASCHK	Alias check. Checks aliases and adds all associated members to the MEMLIST display.
ALL	MEMLIST all. Adds all members in the data set to the MEMLIST display.
ALLOC[ERR]	transits to a tutorial explaining dynamic allocation error codes.
ALTC[MD]	executes a subcommand (previously saved by SETALT) on the current data set.
ALT[ERNAT]	displays an alternate view of the current table
APP[LY]	applies the specified line command to all table entries and executes each entry. Syntax: `APPLY linecmd`
BLK[SIZE]	provides information on disk track utilization for optimal blocking.
CAN[CEL]	cancels pending line commands for the function specified. Syntax: `CAN CSE/CAX/LA/LF/LV/ML/W/ALL`
CI[SIZE]	provides information on optimal control interval sizes for VSAM data sets.
COMM[ENT]	allows any following information to be entered as comments.
DEF[INE]	provides the equivalent of **IDCAMS; DEFINE**.
DR[OP]	terminates an active GO session by number. Syntax: `DROP [*/n] [PROMPT]` (where n defaults to the current GO session)
DUA[L]	displays a double or triple line view of the current table.
EDITL[OG]	(or **EL[OG]**) enters an edit session on the output from the last subcommand.
EDITT[BL]	(or **ET[BL]**) enters an edit session on data from the current function table.
EQ[UATE]	SUBLIST =. Resets the current member group to the members in the current MEMLIST.

GR[OUP] provides a prompt panel for specifying a member group followed by a prompt for a subcommand.

IN[DEX] transits to the STARWARP tutorial index.

LAS[TCMDS] presents the last 32 primary commands for selection by number, for modification and reuse.

LE[FT] rotates through alternate views of the current data. PF keys 10 and 22 are normally set to LEFT.

MC[OPY] interfaces with the extended copy feature using a default data set name as set in MODEL.

M[ENU] provides menu system command and operand entry assistance. You may use MENU to switch between the STARWARP and STARWARP primary panels. Note that items may be chained as in **M.3.3** (or its equivalent, **3.3**). In addition, entry assist panels for most subcommands can be invoked as in **M.FIND** (or its equivalent, **FIND**).
Syntax: `MENU [ON/OFF] [STARTOOL/STARWARP]`

MODEL provides a prompt panel (with initial data set values) for the CREATE or IDCAMS subcommand.

MON[TH] merges members updated or created this month into the MEMLIST.

NOR[MAL] displays the default view of the current table.

O[PTIONS] provides primary command selection for the current function and operand syntax assistance.

OUT[PUT] outputs the current function table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

PAN[EL] selects a panel using ISPF display services (this is normally used for panel testing).
Syntax: `PANEL member`

PEND checks for pending line commands and selects the next one. If no line commands are pending, the command is ignored or the primary MENU is selected if MENU mode is enabled.

PRIM[ER] transits to a STARWARP introductory tutorial.

R[ECALL] (or **RC**) displays the last STARWARP subcommand for modification and/or rentry; RECALL can also retrieve subcommands from the log.

RF[IND] finds a string (repeat find). In the log, it resets the display start location and positions the cursor over the string; otherwise, it just positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the current data. PF keys 11 and 23 are normally set to RIGHT.

SET prompts for one of the following **SET** services.

SETA[LL] controls STARWARP combined defaults

SETALT recalls and saves a subcommand for repeated use later by ALTCMD.

SETC[OLOR] controls screen colors and highlighting.

SETD[SN] sets data set controls.

SETK[EYS] controls PF keys for STARWARP panels (for ISPF 4.x users, the KEYS command may be used for the same effect as SETKEYS).

SETP[ANEL] controls optional panels and panel defaults.

SETSEEK specifies the default member name for the SEEK line command.

SETSEL specifies alias names for S (or SELECT) by function.

SETU[SER] controls dynamic primary commands. You can specify command names and their actions.

SUS[PEND] terminates ISPMODE temporarily and enters line mode processing. ISPMODE may be restarted with the same log table by entering an ISPMODE, ISPXEQ or MEMLIST subcommand. While ISPMODE is suspended, IND$FILE is supported for PC SEND and RECEIVE.

TAG forms a member group containing only members marked with *TAG* in the DATA/MSG field for MEMLIST. Otherwise, applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.

TOD[AY] merges members updated or created today into the MEMLIST.

TRANS provides a translation service to convert a hex, decimal or EBCDIC character to the other forms.

Syntax: `TRANS 1-to-2-hex/1-to-3-decimal/char`
`[HEX/NUMBER/CHAR]`

TRAP interfaces with TSO commands to capture their output lines in the log.

Syntax: `TRAP tsocommand [anyoperands]`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation. Several of these panels also allow dynamic primary commands with which you specify command names and their corresponding actions.

WE[EK] merges members updated or created this week into the MEMLIST.

X clears the current table relative to the cursor position.

Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the current table; this is equivalent to **X ALL**

XC[OPY] interfaces with the extended copy feature.

XM[IT] transmits the data set to another user.

All Commands

The first section of this chapter (see **Common Commands** on page 91) documents ISPMODE only commands which may be entered in any ISPMODE function. This section of this chapter documents all ISPMODE only commands which may be entered (including commands specific to a single function); these are referenced in Index entry "Command".

Global commands are normal STARWARP subcommands which apply to all data sets in a DDNAME/LISTA, LISTC/LISTF or WORKPAD table. A command named GLOBAL is a general global command; you may follow GLOBAL with any STARWARP subcommand and the subcommand will be issued against each data set in the table.

The FIND, REPLACE, MODEL, SEEK and WHOHAS commands are also global for these tables; however, in LISTC/LISTF tables, the FIND, REPLACE, and SEEK commands skip over non-partitioned data sets. This is done because the FIND and REPLACE subcommands use different subcommand syntax for partitioned data sets and non-partitioned data sets. If you actually want to search or update non-partitioned data sets, you may enter **GLOBAL** followed by **FIND** or **REPLACE** and its operands.

The following commands are supported by STARWARP in ISPMODE; several of these commands are supported only in a single function:

***** merges current member group members into the MEMLIST table.

? provides extended help on up to five warning or error messages from the last subcommand.

/* allows any following information to be entered as comments.

ADD in WORKPAD, adds 5 blank table lines to the end of the WORKPAD table.
In LISTV, adds data sets to the LISTF table for all volumes in the LISTV table without prompting.

ALIASCHK Alias check. Checks aliases and adds all associated members to the MEMLIST display.

ALL MEMLIST all. Adds all members in the data set to the MEMLIST display.

ALLOC[ERR] transits to a tutorial explaining dynamic allocation error codes.

ALTC[MD] executes a subcommand (previously saved by SETALT) on the current data set.

ALT[ERNAT] displays an alternate view of the current table

APP[LY] applies the specified line command to all table entries and executes each entry.
Syntax: `APPLY linecmd`

BAT[CHJCL] in DDNAME, LISTA, LOG and MEMLIST builds batch JCL from information in the table.

BLK[SIZE] provides information on disk track utilization for optimal blocking.

CAN[CEL] Cancels pending line commands for the function specified.
Syntax: `CAN CSE/CAX/LA/LF/LV/ML/W/ALL`

CI[SIZE] provides information on optimal control interval sizes for VSAM data sets.

COLS in LOG, provides a column ruler in the log for determining column numbers.

CO[LS] in PBROWSE, displays a columns line on the first line of the data area. The columns line will remain at the top of the data display; it is useful in identifying columns to be used with the FIND command.
Syntax: `COLS [ON/OFF]`

COMM[ENT] allows any following information to be entered as comments.

CONT[INUE] in LOG after a checkpoint, specifies that STARWARP should continue the current interrupted process until the next checkpoint interval as specified by **SETALL**. **END** may be used to terminate the process.

DEF[INE] provides the equivalent of **IDCAMS; DEFINE**.

DR[OP] terminates an active GO session by number.
Syntax: `DROP [*/n] [PROMPT]` (where n defaults to the current GO session)

DUA[L] displays a double or triple line view of the current table.

EDITL[OG] (or **EL[OG]**) enters an edit session on the output from the last subcommand.

EDITT[BL] (or **ET[BL]**) enters an edit session on data from the current function table.

EQ[UATE] SUBLIST =. Resets the current member group to the members in the current MEMLIST.

ER[ASE] in LISTC/LISTF and WORKPAD, deletes the current table in memory and disk.

EXPR[ESS] in CAX, CSECTS, LISTA/DDNAME, LISTC/LISTF, LISTV, MEMLIST, NUCMAP, and WORKPAD, executes all entered line commands without pauses between individual commands.

F finds a string. In LOG, it resets the display start location and positions the cursor over the string; otherwise, it just positions the display start location.
Syntax:
```
F anystring  [ASIS]
       [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
```

FI[ND] global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD, changes to each data set in the table and issues a FIND subcommand. **Note**: in LISTC/LISTF tables, non-partitioned data sets are skipped; in other tables, you should not attempt to search mixed partitioned and non-partitioned data sets with a single FIND global command.
Syntax:
```
FIND memgroup 'anystring'
          [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
          [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
```

F[IND] in PBROWSE, finds a string and positions the display start location.
Syntax:
```
FIND anystring [nn mm] [ASIS]
         [FIRST/LAST/PREV] [PREFIX/SUFFIX/WORD]
```
(where nn and mm are column numbers)

GL[OBAL] global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD, changes to each data set in the table and issues any STARWARP subcommand.
Syntax: `GLOBAL anysubcommand [anyoperands]`

GR[OUP] provides a prompt panel for specifying a member group followed by a prompt for a subcommand.

HEX in PBROWSE, sets HEX display mode on or off.
Syntax: `HEX [ON/OFF]`

ID in LISTC/LISTF and WORKPAD, changes the table ID name.
Syntax: `ID name` (where name is a one to six character alphameric name).

IN[DEX] transits to the STARWARP tutorial index.

INS[ERT] in WORKPAD, adds blank WORKPAD table lines.
Syntax: `INSERT [nn] [TOP/BOTTOM/HERE]` (where nn defaults to 5)

LAS[TCMDS] presents the last 32 primary commands for selection by number, for modification and reuse.

LE[FT] rotates through alternate views of the current data. PF keys 10 and 22 are normally set to LEFT.

LISTVT[OC] in LISTV, adds data sets to the LISTF table for all volumes in the LISTV table after a prompt.

LO[AD] in LISTC/LISTF and WORKPAD, loads data set names from operating system control blocks.
Syntax: `LOAD APFLST/LNKLST/LPALST  [RESET]`

L[OCATE] in CAX, CSECTS, LISTC/LISTF, LISTV, MEMLIST, and NUCMAP, positions to a data line in sorted tables by searching the current sorted column for the specified data. LOCATE may be entered without operands to go to a prompting panel.
Syntax: `LOCATE datavalue`

L[OCATE] in PBROWSE, positions directly to the specified record number.
Syntax: `LOCATE recnumber`

MC[OPY] interfaces with the extended copy feature using a default data set name as set in MODEL.

M[ENU] provides menu system command and operand entry assistance. You may use MENU to switch between the STARWARP and STARWARP primary panels. Note that items may be chained as in **M.3.3** (or its equivalent, **3.3**). In addition, entry assist panels for most subcommands can be invoked as in **M.FIND** (or its equivalent, **FIND**).
Syntax: `MENU [ON/OFF] [STARTOOL/STARWARP]`

MER[GE] in MEMLIST, LISTC/LISTF and WORKPAD, merges data from a different table into the current table.
Syntax: `MERGE name [GROUP/SAMPLE] [RESET]`
(where name is one to six alphameric characters)
`[TOP/BOTTOM/HERE]` (for WORKPAD only)

MINE in MEMLIST, builds a member list of members with ISPF statistics which were saved by the current USERID.

MODE in WORKPAD, specifies if modified WORKPAD lines are to be automatically executed.
Syntax: `MODE [EXEC/NOEXEC]`

MODEL global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD, changes to each data set in the table and issues a MODEL command.
Otherwise, provides a entry assist panel (with initial data set values) for the CREATE or IDCAMS subcommand.

MON[TH] merges members updated or created this month into the MEMLIST.

NO in LOG after a reply required, specifies that STARWARP should not complete the current action. A **YES** response would allow the process to continue.

NOR[MAL] displays the default view of the current table.

NOS[AVE] in LISTC/LISTF and WORKPAD, specifies that the current table should not be saved on disk regardless of the setting of "Automatic save" in SETALL for LISTC/LISTF or WORKPAD.

O[PTIONS] provides primary command selection for the current function and operand syntax assistance.

OUT[PUT] outputs the current function table to print or a data set.
Syntax: `OUTPUT [=c / F(ddname)]`

PAN[EL] selects a panel using ISPF display services (this is normally used for panel testing).
Syntax: `PANEL member`

PAD in PBROWSE, changes the padding character and specifies the number of pad characters desired.
Syntax: `PAD character/'character'/x'hex'  [column]`
Example: PAD a 45

PEND checks for pending line commands and selects the next one. If no line commands are pending, the command is ignored or the primary MENU is selected if MENU mode is enabled.

PRIM[ER] transits to a STARWARP introductory tutorial.

RCH[ANGE] in LOG, captures the command or data set name under the cursor in the log for modification and/or reuse. PF keys 6 and 18 are normally set to RCHANGE.

R[ECALL] (or **RC**) displays the last STARWARP subcommand for modification and/or rentry; RECALL can also retrieve subcommands from the log.

REF[RESH] global command in LISTC/LISTF and LISTV, updates information and status for all lines in the table.

REM[OVE] in CAX, CSECTS, LISTA/DDNAME, LISTC/LISTF, LISTV, MEMLIST, NUCMAP and WORKPAD, trims a dialog table based on a string match.

```
Syntax: REMOVE anystring [columname/ALL]
               [PREFIX/SUFFIX/WORD] [NOT]
```

REPL[ACE] global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD, changes to each data set in the table and issues a REPLACE subcommand. **Note**: in LISTC/LISTF tables, non-partitioned data sets are skipped; in other tables, you should not attempt to update mixed partitioned and non-partitioned data sets with a single REPLACE global command.

```
Syntax: REPLACE memgroup 'fromstring' 'tostring'
               [NUM/SNUM/NONUM/LBLOCK/LDUMP/BLOCK/DUMP]
               [CAPS/ASIS/IGNORE] [WORD/PREFIX/SUFFIX]
               [WRITE/NOWRITE]
```

RESET in DDNAME, LISTA, LISTC, LISTF and WORKPAD, often used with data set tagging to clear the DATA/MSG field in all table entries.

RF[IND] in PBROWSE, finds a string (repeat find) and positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RF[IND] finds a string (repeat find). In the log, it resets the display start location and positions the cursor over the string; otherwise, it just positions the display start location. PF keys 5 and 17 are normally set to RFIND.

RI[GHT] rotates through alternate views of the current data. PF keys 11 and 23 are normally set to RIGHT.

SAM[PLE] in WORKPAD, merges table number 99 from the installation table library (usually ISPTLIB) into the current WORKPAD table. Table 99 is an index to other sample WORKPAD tables. The additional sample tables can be included by using the MERGE command.

SAVE in MEMLIST, LISTC/LISTF and WORKPAD, creates a permanent table for use in a later STARWARP session.

Syntax: `SAVE [name] [REPLACE/NOREPL]`

(where name is one to six alphameric characters)

SEEK global command in LISTA/DDNAME, LISTC/LISTF and WORKPAD, changes to each data set in the table and looks for a member. Note: in LISTC/LISTF, non-partitioned data sets are skipped.

Syntax: `SEEK member`

S[ELECT] in MEMLIST and LOG, normally BROWSEs load members and EDITs source members. The value used for the command is set in SETSEL.

Syntax: `SELECT member`

SET prompts for one of the following **SET** services.

SETA[LL] controls STARWARP combined defaults

SETALT recalls and saves a subcommand for repeated use later by ALTCMD.

SETC[OLOR] controls screen colors and highlighting.

SETD[SN] sets data set controls.

SETK[EYS] controls PF keys for STARWARP panels (for ISPF 4.x users, the KEYS command may be used for the same effect as SETKEYS).

SETP[ANEL] controls optional panels and panel defaults.

SETSEEK specifies the default member name for the SEEK line command.

SETSEL specifies alias names for S (or SELECT) by function.

SETU[SER] controls dynamic primary commands. You specify command names and actions.

SO[RT] in CAX, CSECTS, LISTC/LISTF, LISTV, MEMLIST or NUCMAP, sorts function tables into an alternate order. SORT with no operands sorts the table in default order, while SORT with a field name sorts the table in that order. If you enter an invalid sort command such as **SORT xx**, a prompt will show the valid sort fields for that function. Syntax: `SORT [name] [ASCEND/DESCEND]`

SPA[CE] in LISTV, updates volume space statistics for each volume in the table.

STATS in LISTV, updates volume attributes for each volume in the table.

SUS[PEND] terminates ISPMODE temporarily and enters line mode processing. ISPMODE may be restarted with the same log table by entering an ISPMODE, ISPXEQ or

MEMLIST subcommand. While ISPMODE is suspended, IND$FILE is supported for PC SEND and RECEIVE.

TAG in MEMLIST, forms a member group containing only members marked with *TAG* in the DATA/MSG field. Otherwise, applies the specified line command to table entries marked with *TAG* in the DATA/MSG field and executes each entry.
Syntax: `TAG linecmd`

TOD[AY] merges members updated or created today into the MEMLIST.

TRANS provides a translation service to convert a hex, decimal or EBCDIC character to the other forms.
Syntax: `TRANS {1-to-2-hex/1-to-3-decimal/char}`
`[HEX/NUMBER/CHAR]`

TRAP interfaces with TSO commands to capture their output lines in the log.
Syntax: `TRAP tsocommand [anyoperands]`

UT selects the extended user command panel. Commands are maintained in these panels by the STARWARP developers as well as your installation. Several of these panels also allow dynamic primary commands with which you specify command names and their corresponding actions.

WE[EK] merges members updated or created this week into the MEMLIST.

WHO[HAS] global command in LISTC/LISTF and WORKPAD, issues a WHOHAS subcommand for each data set in the table.

X clears the current table relative to the cursor position.
Syntax: `X [ABOVE/BELOW/ALL]`

XA[LL] clears the current table; this is equivalent to **X ALL**

XC[OPY] interfaces with the extended copy feature.

XM[IT] transmits the data set to another user.

Y[ES] in LOG, after a reply required, specifies that STARWARP should continue the current action. A **NO** response would stop the current action.

Appendix A. Member Name Forms

STARWARP allows many different member name forms. Member names may be entered in either character or hexadecimal mode for any subcommand which processes member names.

Character member names may be from one to eight bytes long with no imbedded blanks, commas, parentheses, colons, slashes, asterisks, question marks or percent symbols.

Hexadecimal member names may contain from one to sixteen hexadecimal digits delimited by **x'** and **'**. **X'd7c4e2c5'** and **PDSE** are entirely equivalent; also, **x'333'** and **x'0333'** are equivalent.

Subcommands ATTRIB, BROWSE, COMPDIR, DCF, DELINK, DIRENTRY, DISASM, EDIT, EXCLUDE, FIND, FSE, HISTORY, IF, LIST, MAP, MEMBERS, MEMLIST, PGMDOC, PRINT, READOBJ, REPLACE, REVIEW, SPFEDIT, SUBLIST, SUBMIT, TSOEDIT, TSOLIST, VERIFY, VPRINT and XREF allow additional member name specifications.

A member name range, member name pattern or member name combination forms what is termed a member group. A simple member name, a pattern or a combination name may contain ? or % characters as placeholders. Note that a member list (that is, a parenthesized list containing member and member group specifications separated by commas) may also be used.

When a member group is being processed, the subcommand processing routine is re-executed for each member in the group. Specifically, note the following:

Default member name if no member name is specified, the last member name or member group entered for any of the above subcommands will be assumed.

Asterisk member name if ***** is entered in the member name position, the last member name or member group entered for any of the above subcommands will be assumed.

Equal member name if **=** is entered in the member name position, the list of members in the current MEMLIST table will be assumed. If no MEMLIST table is active, the current member group is assumed (as for *).

Member name range any of these subcommands may select all members in a range by specifying a beginning partial member name, a colon, and a ending partial member name (either or both of the partial names may be null).

Logic from the DISPLAY subcommand is used to select members; the subcommand processes these selected members in alphabetic order. The following specifications would select the same member names:

```
subcommand :          and  DISPLAY
subcommand part1:     and  DISPLAY part1
subcommand :part2     and  DISPLAY x'00' part2
subcommand prt1:prt2  and  DISPLAY prt1 prt2
```

Member name pattern any of these subcommands may select all members whose names contain pattern characters by specifying a member name segment, a slash, and another member name segment (either, not both, of the member name segments may be null).

Logic from the PATTERN subcommand is used to select members; the subcommand processes these selected members in alphabetic order. The following specifications would select the same member names:

```
subcommand seg1/      and  PATTERN seg1
subcommand /seg2      and  PATTERN seg2
subcommand seg1/seg2  and  PATTERN seg1 seg2
```

Member name combination any of these subcommands may select all members whose names contain a range of characters and a pattern of characters by specifying a member name segment, an asterisk and another member name segment (either, not both, of the member name segments may be null).

Logic from DISPLAY and PATTERN subcommand is used to select members; the subcommand processes these selected members in alphabetic order. The following specifications would select the same member names:

```
subcommand seg1*      and  DISPLAY seg1 seg1
subcommand *seg2      and  PATTERN seg2
subcommand seg1*seg2  and  DISPLAY seg1 seg1 --with-- PATTERN seg2
```

Member name placeholder a simple member name, a member name pattern or a member name combination may contain a ? or % character as a "wild card" character in any position.

Logic from DISPLAY and PATTERN subcommand is used to select members; the subcommand processes these selected members in alphabetic order. The following specifications would select the same member names:

```
subcommand ab?       and  DISPLAY x'c1c20040' x'c1c2ff40'
subcommand ab?*      and  DISPLAY x'c1c200'   x'c1c2ff'
subcommand *a?c      and  PATTERN a?c
subcommand /a%c      and  PATTERN a%c
subcommand a%c/xx    and  PATTERN a%c xx
```

Member group examples The following examples each define and display a member group:

```
SUBCOMMAND         MEMBERS IN THE GROUP
----------         --------------------
MEMBERS aa         AA
MEMBERS (aa,bb)    AA and BB
MEMBERS (aa,bb/)   AA and members whose names contain BB
MEMBERS *          current member group
MEMBERS            current member group
MEMBERS =          members in the current MEMLIST

MEMBERS :          all members -- X'00' through X'FF'
MEMBERS aa:        members from AA... through X'FF'
MEMBERS :bb        members from X'00' through BB...
MEMBERS aa:bb      members between AA... and BB...
MEMBERS (abc,d:)   member ABC and those from D... through X'FF'

MEMBERS aa/        members whose names contain AA anywhere
MEMBERS /bb        members whose names contain BB anywhere
MEMBERS aa/bb      members whose names contain AA and BB
MEMBERS (aa/,bb/)  members whose names contain AA or BB

MEMBERS aa*        members with names AA...
MEMBERS *bb        members whose names contain BB anywhere
MEMBERS aa*bb      members with names AA... and BB elsewhere
MEMBERS (aa*,bb/)  members with names AA... or with BB anywhere

MEMBERS aa?        members with names AA. (three characters long)
MEMBERS b?b        members with names B.B (three characters long)
MEMBERS %a*b?d     members with names .A and B.D after that
MEMBERS ?a/b%d     members with names .A and B.D in the name
MEMBERS (aa?,?b/)  members with names AA. or with .B anywhere
```

Member name displays subcommands MEMBERS, PATTERN and DISPLAY allow the display of member groups using the above member group syntax; these subcommands differ as follows:

DISPLAY can not modify the current member group; if no member specification is entered, the entire member directory is displayed. Member list notation may not be used.

PATTERN can not modify the current member group; if no member specification is entered, the member group last entered on a PATTERN subcommand is displayed. Member list notation may not be used.

MEMBERS can modify the current member group; if no member specification is entered, the current member group is displayed. Note: MEMBERS is one of the subcommands which can define a member group and member list notation may be used.

Appendix B. ABEND Processing

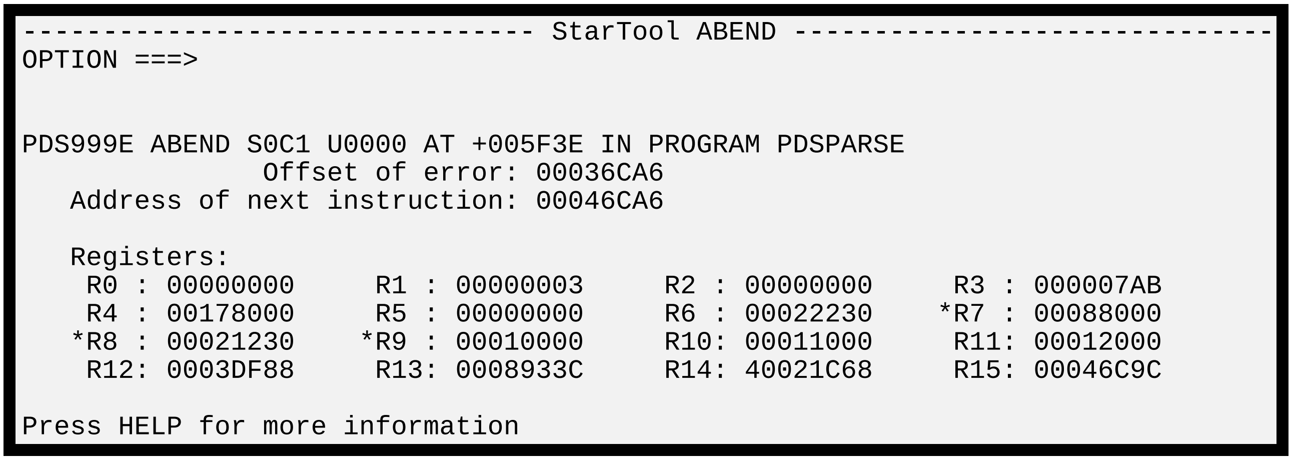
STARWARP always attempts to recover from ABENDS with its ESTAE and ESTAI recovery routines unless CONTROL NORECOVER is in effect. STARWARP performs ABEND processing as follows:

1. The STARWARP program uses different recovery methods depending on the subcommand being executed. The method used should perform any cleanup and reinitialization functions required to restart STARWARP.
   a) For internal STARWARP subcommands, recovery is performed by writing an error message and terminating the subcommand.
   b) For the external subcommands BROWSE, EDIT, or ISPF, no ESTAE recovery is attempted; instead normal ISPF processing is permitted.
   c) For the other external STARWARP subcommands (COMPARE, DCF, EXEC, FSE, HELP, PRINT, REVIEW, SUBMIT, TSO xxx, TSOEDIT, TSOLIST and VPRINT), the subcommand processor is forcibly detached.
2. If CONTROL NORECOVER is in effect, STARWARP will terminate unless the subcommand being used specifically tests for ABEND conditions (such as IF with LOADERR or VERIFY with LOAD).
3. If CONTROL RECOVER is in effect, any ABEND dump will probably not be very useful; also, if you want to use TSO TEST after an ABEND, it is recommended that you set CONTROL NORECOVER first.
4. Certain external programs reached via the TSO subcommand (such as CALL) will terminate STARWARP with a "READY" message if the program they invoke ABENDS. Note, however, that if a null line is entered at this point, STARWARP will regain control.
5. The PDS999E message should identify the name of the STARWARP assembly listing involved in the error as follows:

PDSMAIN	ABEND in the STARWARP mainline
PDS#SECI	ABEND in the security interface
PDSALIAS	ABEND in the subroutine assembly
PDSCBSX	ABEND in the Copybook setup routine
PDSCPARS	ABEND in the Copybook parse routine
PDSDECOD	ABEND in the DISASM routine
PDSDELNK	ABEND in the DELINK routine
PDSFCALC	ABEND in the CALC routine
PDSFILE	ABEND in the STARWARP batch execution interface
PDSFPARS	ABEND in the STARWARP batch emulation parser
PDSIDCAM	ABEND in the IDCAMS interface
PDSIDSPY	ABEND in the ISPMODE dialog
PDSIPARS	ABEND in the STARWARP parser
PDSPARSE	ABEND in the PARSE interface
PDSSPACE	ABEND in the service routines
PDSVTOCR	ABEND in the VTOC read routine
VTSOCMD	ABEND in the TSO command check

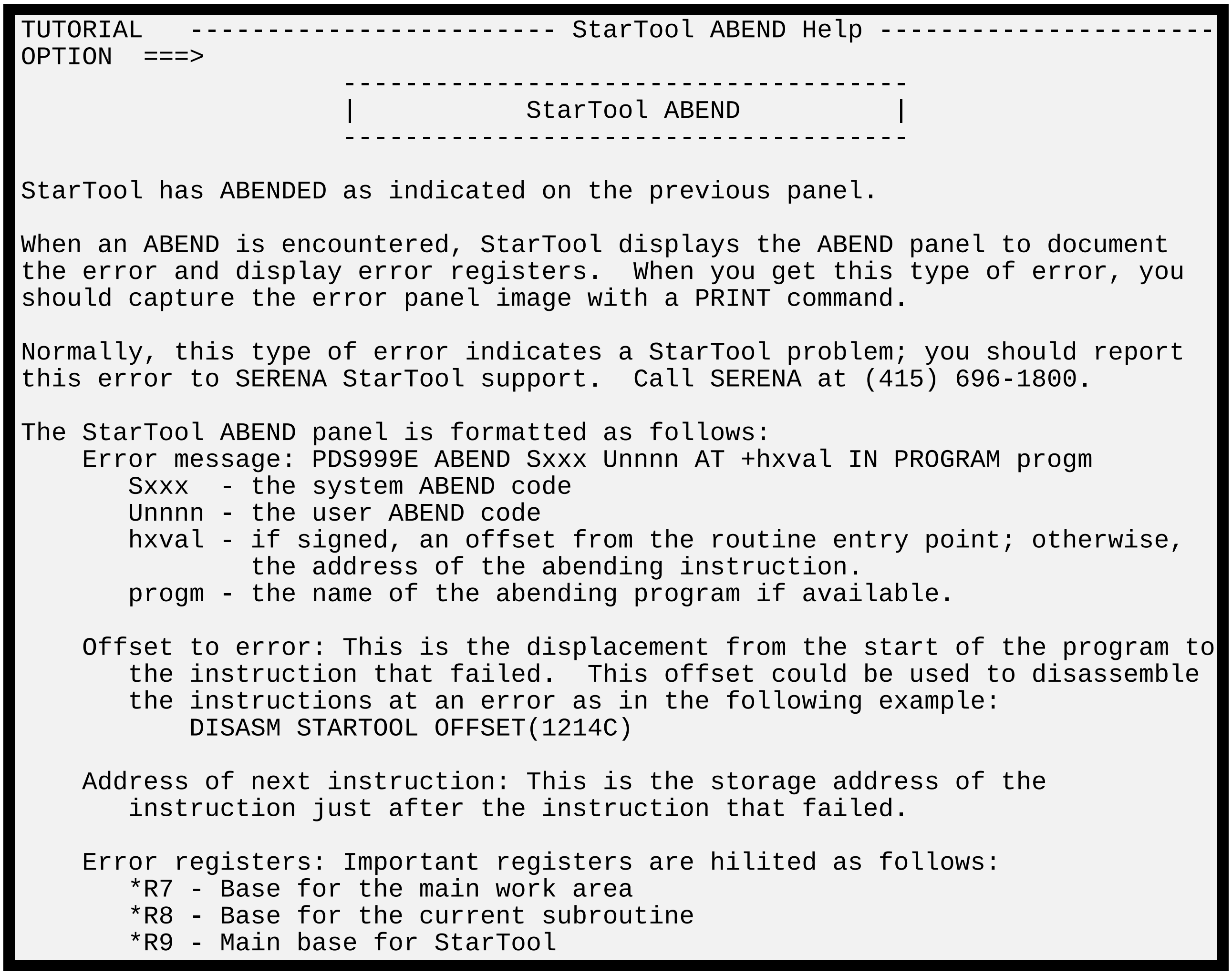
If you need assistance with STARWARP, contact your marketing representative. If you desire futher assistance, contact **SERENA**.

Appendix B. ABEND Processing



```
-------------------------------- StarTool ABEND ------------------------------
OPTION ===>


PDS999E ABEND S0C1 U0000 AT +005F3E IN PROGRAM PDSPARSE
               Offset of error: 00036CA6
   Address of next instruction: 00046CA6

   Registers:
    R0 : 00000000     R1 : 00000003     R2 : 00000000     R3 : 000007AB
    R4 : 00178000     R5 : 00000000     R6 : 00022230    *R7 : 00088000
   *R8 : 00021230    *R9 : 00010000     R10: 00011000     R11: 00012000
    R12: 0003DF88     R13: 0008933C     R14: 40021C68     R15: 00046C9C

Press HELP for more information
```

Figure 180. Sample ABEND panel



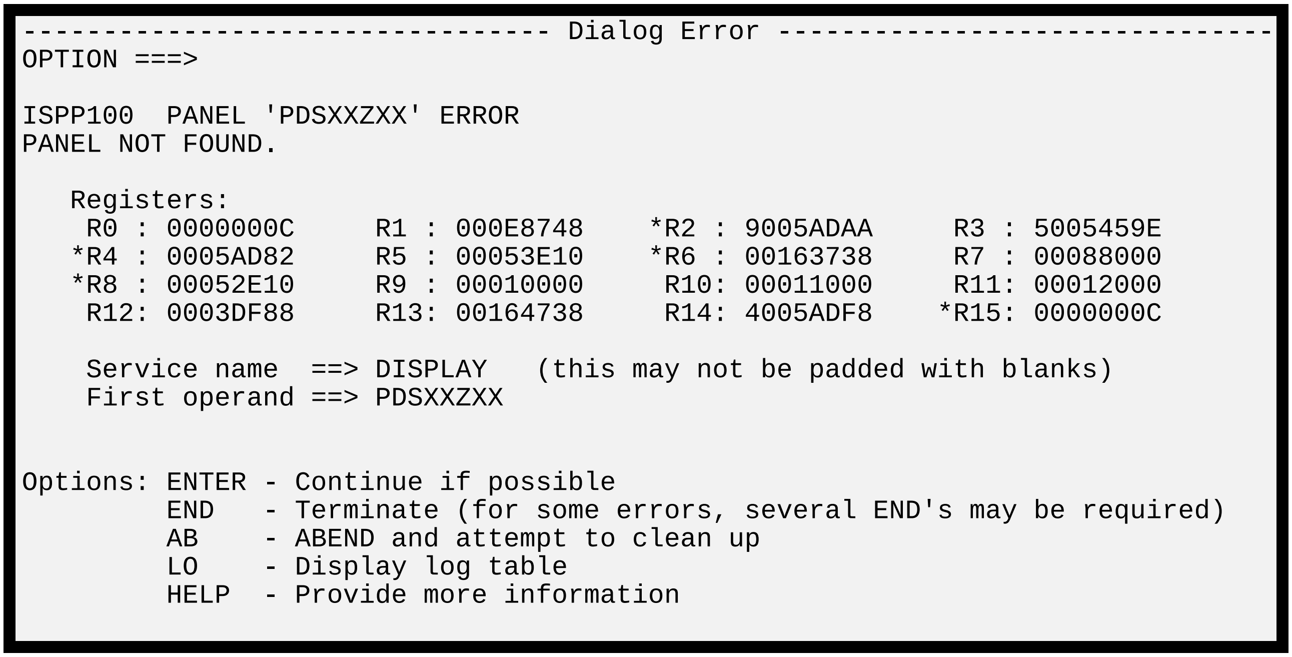
```
TUTORIAL   ------------------------ StarTool ABEND Help ----------------------
OPTION  ===>
                     -------------------------------------
                     |           StarTool ABEND          |
                     -------------------------------------

StarTool has ABENDED as indicated on the previous panel.

When an ABEND is encountered, StarTool displays the ABEND panel to document
the error and display error registers.  When you get this type of error, you
should capture the error panel image with a PRINT command.

Normally, this type of error indicates a StarTool problem; you should report
this error to SERENA StarTool support.  Call SERENA at (415) 696-1800.

The StarTool ABEND panel is formatted as follows:
    Error message: PDS999E ABEND Sxxx Unnnn AT +hxval IN PROGRAM progm
       Sxxx  - the system ABEND code
       Unnnn - the user ABEND code
       hxval - if signed, an offset from the routine entry point; otherwise,
               the address of the abending instruction.
       progm - the name of the abending program if available.

    Offset to error: This is the displacement from the start of the program to
       the instruction that failed.  This offset could be used to disassemble
       the instructions at an error as in the following example:
           DISASM STARTOOL OFFSET(1214C)

    Address of next instruction: This is the storage address of the
       instruction just after the instruction that failed.

    Error registers: Important registers are hilited as follows:
       *R7 - Base for the main work area
       *R8 - Base for the current subroutine
       *R9 - Main base for StarTool
```

Figure 181. Sample ABEND help panel

Appendix C. Dialog Errors

STARWARP monitors for dialog errors and displays a special panel when an ISPF dialog error is encountered. This panel presents the user with recovery options, formats an error message and hilites important registers.

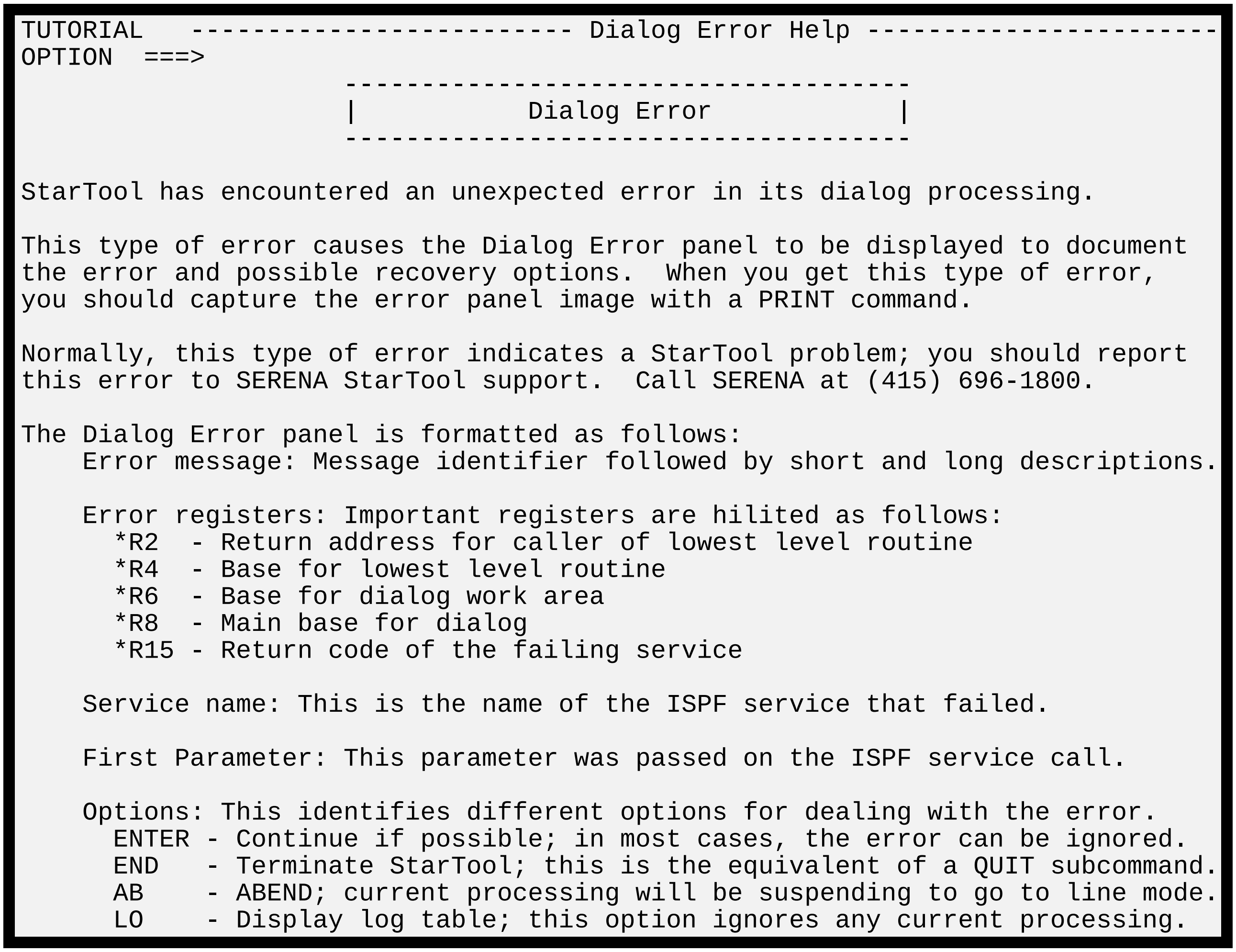
If you need assistance with STARWARP, contact your marketing representative. If you desire futher assistance, contact **SERENA**.



```
--------------------------------- Dialog Error -------------------------------
OPTION ===>

ISPP100  PANEL 'PDSXXZXX' ERROR
PANEL NOT FOUND.

   Registers:
    R0 : 0000000C     R1 : 000E8748    *R2 : 9005ADAA     R3 : 5005459E
   *R4 : 0005AD82     R5 : 00053E10    *R6 : 00163738     R7 : 00088000
   *R8 : 00052E10     R9 : 00010000     R10: 00011000     R11: 00012000
    R12: 0003DF88     R13: 00164738     R14: 4005ADF8    *R15: 0000000C

    Service name  ==> DISPLAY   (this may not be padded with blanks)
    First operand ==> PDSXXZXX


Options: ENTER - Continue if possible
         END   - Terminate (for some errors, several END's may be required)
         AB    - ABEND and attempt to clean up
         LO    - Display log table
         HELP  - Provide more information
```

Figure 182. Sample Dialog Error Panel

Appendix C. Dialog Errors



```
TUTORIAL   ------------------------- Dialog Error Help -----------------------
OPTION  ===>
                     -------------------------------------
                     |           Dialog Error           |
                     -------------------------------------

StarTool has encountered an unexpected error in its dialog processing.

This type of error causes the Dialog Error panel to be displayed to document
the error and possible recovery options.  When you get this type of error,
you should capture the error panel image with a PRINT command.

Normally, this type of error indicates a StarTool problem; you should report
this error to SERENA StarTool support.  Call SERENA at (415) 696-1800.

The Dialog Error panel is formatted as follows:
    Error message: Message identifier followed by short and long descriptions.

    Error registers: Important registers are hilited as follows:
      *R2  - Return address for caller of lowest level routine
      *R4  - Base for lowest level routine
      *R6  - Base for dialog work area
      *R8  - Main base for dialog
      *R15 - Return code of the failing service

    Service name: This is the name of the ISPF service that failed.

    First Parameter: This parameter was passed on the ISPF service call.

    Options: This identifies different options for dealing with the error.
      ENTER - Continue if possible; in most cases, the error can be ignored.
      END   - Terminate StarTool; this is the equivalent of a QUIT subcommand.
      AB    - ABEND; current processing will be suspending to go to line mode.
      LO    - Display log table; this option ignores any current processing.
```

Figure 183. Sample Dialog Error Help Panel

Appendix D. Attention Processing

When STARWARP is executed as an ISPF dialog, attentions are not normally required since an output loop will be detected by the checkpoint processing of ISPMODE. If STARWARP or a supporting TSO command is in a CPU loop (with no output), you need to interrupt the process with an attention key or it will continue indefinitely.

In general, an attention (the PA1 key) may be used to terminate the current STARWARP subcommand and two attentions in a row may be used to terminate the STARWARP command. STARWARP performs attention processing as follows:

1. A single (double for TSOEDIT) attention should terminate any member group in progress (except for the FSE subcommand).
2. An attention at a prompt for a data set should terminate STARWARP.
3. An attention at the normal PDS300A ENTER OPTION subcommand prompt should provide another ENTER OPTION prompt; another attention at this point should terminate STARWARP.
4. For subcommands BROWSE, EDIT, ISPF, ISPMODE, or MEMLIST, no ENTER OPTION prompt is provided by STARWARP's attention handling:
   a) A single attention should terminate any member group in progress.
   b) An attention during a edit or browse line command from MEMLIST will terminate any remaining line commands.
   c) Any additional attentions should be ignored by STARWARP.
   d) For TSO commands invoked by ISPF with no attention handling: the first attention is noted by STARWARP and it will terminate any member group in progress; subsequent attentions apply only to the TSO command.
5. For other STARWARP subcommands, a single (double for TSOEDIT) attention should result in a PDS300A ENTER OPTION prompt; the processing action taken at this point depends on what is entered next:
   a) Another attention should terminate the STARWARP command.
   b) A null line should cause the interrupted subcommand to continue at the point of interruption. Note: any messages awaiting output at the time of the interrupt will have already been discarded.
   c) If a subcommand is entered and the currently executing subcommand is external to STARWARP (COMPARE, DCF, EXEC, FSE, HELP, PRINT, REVIEW, SUBMIT, TSO xxx, TSOEDIT, TSOLIST or VPRINT), the subcommand processor should be forcibly detached and the ABEND message should be suppressed.
   d) An entered subcommand should be given control without delay.
6. If an internal STARWARP subcommand is in a loop that does not involve input or output to the terminal or input from the current data set, a single attention will not terminate the loop:
   a) The first attention will provide the normal PDS300A ENTER OPTION prompt.
   b) If a subcommand is entered at this time, STARWARP will continue looping.
   c) Another attention will produce the message "PDS470W The program is probably in a loop" and STARWARP will perform the subcommand entered at the ENTER OPTION prompt.

If you need assistance with STARWARP, contact your marketing representative. If you desire further assistance, contact **SERENA**.

Appendix E. Update Protection

Several STARWARP subcommands can modify data sets when they are allocated as "shared".

If the data set is allocated as OLD, the STOW DCB is left open after the first data set update. For OLD allocations, exclusive use of the data set is assured and the following RESERVE logic does not apply.

If the data set is allocated as SHR (SHR was specifically requested or SHR was defaulted to), STARWARP subcommands which update the data set such as ATTRIB (with attributes to be changed) or VERIFY (for update member PDS tests) could cause data set integrity problems if the data set is also being updated by another user.

To circumvent this problem, STARWARP uses the following ISPF RESERVE logic to maintain data set integrity during data set updates.

1. A **RESERVE (SPFEDIT,DSNAME,E,44,SYSTEMS),UCB=ADDRESS** is performed.
2. If the data set is a load library, then a linkage editor RESERVE or ENQUE is performed as appropriate:
   a) On a shared DASDI volume,
   RESERVE (SYSIEWLP,DSNAME,E,44,SYSTEMS),UCB=ADDRESS
   b) On a non-shared DASDI volume,
   ENQ (SYSIEWLP,DSNAME,E,44,SYSTEM)
3. The data set STOW DCB is opened
4. All data set updates are performed.
   a) Wherever possible, the STOW DCB is left open while an entire member group is being processed.
   b) For executions of STARWARP in batch mode, the volume reserve is maintained and the STOW DCB is left open following the first update to the data set.
5. The data set STOW DCB is closed
6. DEQ's are performed to remove the RESERVE's and ENQ's

GLOSSARY

Alias member

A member whose directory entry has an indicator value which indicates that the member entry is an alternate name for a main member; a main member and its aliases are associated with the same recorded data in a data set.

Apparent alias member

Data in a PDS which is pointed to by more than one main directory entry.

Associated members

Members with the same TTR address. For a main member, the associated members could be alias members or an apparent alias member; for an alias member, the associated members could be other aliases or a main member.

Batch mode

A mode of STARWARP operation. In batch mode, all user communication is performed with PUTGET, GETLINE and PUTLINE TSO services. ISPF services are available. In this mode of operation, STARWARP operates under control of the batch Terminal Monitor Program (the TMP, or IKJEFT01).

BLDL

A BPAM function which is used to check for the existence of members.

BPAM

An acronym for Basic Partitioned Access Method; BPAM is actually very similar in use to BSAM which is for sequential data. STARWARP often uses BPAM for member check operations and always uses BPAM for directory updates and reading PDSE data sets.

BSAM

An acronym for Basic Sequential Access Method; STARWARP uses BSAM for several utility functions.

CCHHR

An acronym for Cylinder, Cylinder, Head, Head, Record which is a ten byte hexadecimal disk address relative to the start of the volume.

Command

A request for an operation. For example, STARWARP is a command and FIXPDS is a STARWARP subcommand. The distinction between a command and a subcommand is not often important.

Compress

A process which removes deleted members from a PDS. Disk space formerly occupied by these deleted members is made available for new members. In a PDSE data set, space from deleted members is managed dynamically; you can not compress a PDSE.

Deleted member

Member data which is not pointed to by any directory entry; deleted members remain in a PDS until the data set is compressed. In a PDSE data set, space from deleted members is managed dynamically; you can not restore deleted members.

Directory

A segment of a PDS which is a sequential data set with eight byte keys and 256 byte data records containing pointers to data in the PDS member portion of the data set. For a PDSE, PDS directory and member data is created in a virtual PDS data set.

Directory entry

A logical entry in a PDS or PDSE directory which points to its associated data in the member data; it consists of an eight character member name field and one or more three byte relative address (TTR) fields.

DS1LSTAR

A pointer in the data set control block (DSCB) which indicates the last used disk address for a data set. This pointer is updated after a member is added to a PDS data set or after a PDS is compressed.

Glossary

EXCP

An acronym for EXecute Channel Program. This access method is used extensively by STARWARP to read an entire disk track with a single operation.

Function

A request for an ISPF (or ISPMODE) operation. Most ISPMODE functions support their own commands.

ISPMODE

A mode of STARWARP operation. With ISPMODE, STARWARP operates as an ISPF dialog and normal ISPF services (HELP, TSO, SPLIT, SWAP,...) as well as all STARWARP services are available.

Line command

A command entered on a table line in the CMD field. Line commands may be up to four characters long.

Line mode

A mode of STARWARP operation. In line mode, all user communication is performed with PUTGET, GETLINE and PUTLINE TSO services. No ISPF services are available.

Load library

A PDS or PDSE which has record format U and is commonly used to contain executable modules.

Main member

A non-alias directory entry and its associated recorded data.

Member

Data in a PDS or PDSE which is pointed to by one or more directory entries. Members may be alias or main members. Deleted members which have no associated directory entries. For a PDSE, space from deleted members is managed dynamically; you can not restore deleted members or compress a PDSE.

Module

A member contained in a load library. Also known as a load module or load member.

Orphan member

A member marked as an alias which does not have an associated main member.

PDS or Partitioned data set

PDS. A data set with fixed, variable or undefined format which is commonly used as a *library* for related types of information. Partitioned data sets contain two segments of data: a directory of information and member data. This type of library is requested as DSNTYPE(PDS) in JCL or a TSO ALLOCATE.

PDSE or Partitioned Data Set Extended

PDSE. A SMS managed data set which is logically similar to a PDS. This type of library is requested as DSNTYPE(LIBRARY) in JCL or on an ALLOCATE command. A PDSE can be accessed by BPAM or BSAM access methods to obtain a virtual data set which looks like a PDS.

Primary command

A subcommand or command entered from the command line of a panel.

QSAM

An acronym for Queued Sequential Access Method; STARWARP uses QSAM for several utility functions.

Source library

A PDS or PDSE data set which has fixed or variable format data; source libraries are normally used to contain non-executable data.

STARWARP

The premier year 2000 workbench for MVS environments. A multipurpose ISPF dialog and TSO command processor which manipulates data sets, members and data records.

Subcommand

A request for an operation that is within the scope of work requested by the previously issued command. STARWARP is a command; FIXPDS is a STARWARP subcommand. The distinction between a subcommand and a command is not often important.

TTR

An acronym for Track, Track, Record (a 1 to 6 digit hexadecimal disk address relative to the start of the data set); this type of address is stored in the member directory entry to indicate the start of a member.

Index